```
/* Minify CSS output using Minifier 1.3.5 on 2025-08-29 */
.mw-parser-output .navbar{display:inline-block;margin-left:0.5em;font-weight:normal;font-size:0.8125em;line-height:1.5;text-transform:uppercase;letter-spacing:0.0625em;vertical-text-alignment:baseline;}
.mw-parser-output .navbar-collapse{float:left;}
.mw-parser-output .navbar-header{display:none;}
.mw-parser-output .navbar-ul{display:inline-block;white-space:nowrap;}
.mw-parser-output .navbar-li{display:inline-block;}
.mw-parser-output .navbar-text{display:inline-block;}
.mw-parser-output .navbar-brackets{display:inline-block;}
.mw-parser-output .navbar-mini{display:inline-block;}
.mw-parser-output .navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
.mw-parser-output .navbar-ul.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini.navbar-mini-text.navbar-mini{display:inline-block;}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS 0.91//EN" "http://www.netscape.com/publish/formats/rss/rss-0.91.dtd">
<rss version="0.91">
<channel>
<title>Slashdot: Stories</title>
<link>http://slashdot.org/</link>
<description>Stuff that matters.</description>
<language>en-us</language>
<item>
<title>Ex-Google Employee Sues Over Age Discrimination</title>
<link>http://rss.slashdot.org/~r/Slashdot/slashdot/~3/Ex-Google-Employee-Sues-Over-Age-Discrimination/0001000000000000000000</link>
<description>A former employee of Google working on the Google Translate project has filed a lawsuit claiming he was fired because of his age. Lawrence Hoffman, who was 55 when he was let go from his position as a software engineer, is suing the company for $1.5 million in lost wages and other damages. Hoffman's lawsuit also seeks to have Google change its employment practices, and to have the company pay the legal fees of Hoffman's lawyers. Hoffman's lawsuit was filed in California Superior Court in San Mateo County on Tuesday. Hoffman's lawyer, Lawrence Organ, says that Hoffman was let go from his position in 2009 and was replaced by a 27-year-old. Organ says that Hoffman was "the oldest person in the department." Hoffman's lawsuit also claims that Google's "employment practices have a disparate impact on its older employees
```

```
// load in the main lua file
luaL_dofile(L, "Resources/Scripts/main.lua");if (luaL_dostring(L, "if (type(Initialize) == 'function') Initialize()"
    ".. [[Error running function 'Initialize': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(Update) == 'function') Update(0.0)"
    ".. [[Error running function 'Update': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(Draw) == 'function') Draw()"
    ".. [[Error running function 'Draw': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnMouseButtonPressed) == 'function') OnMouseButtonPressed(0,0,0,0)"
    ".. [[Error running function 'OnMouseButtonPressed': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnMouseButtonReleased) == 'function') OnMouseButtonReleased(0,0,0,0)"
    ".. [[Error running function 'OnMouseButtonReleased': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnMouseMove) == 'function') OnMouseMove(0,0,0,0)"
    ".. [[Error running function 'OnMouseMove': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnMouseWheel) == 'function') OnMouseWheel(0,0,0,0)"
    ".. [[Error running function 'OnMouseWheel': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnKeyPressed) == 'function') OnKeyPressed(0,0,0,0)"
    ".. [[Error running function 'OnKeyPressed': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnKeyReleased) == 'function') OnKeyReleased(0,0,0,0)"
    ".. [[Error running function 'OnKeyReleased': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTextInput) == 'function') OnTextInput(0,0,0,0)"
    ".. [[Error running function 'OnTextInput': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchBegin) == 'function') OnTouchBegin(0,0,0,0)"
    ".. [[Error running function 'OnTouchBegin': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchEnd) == 'function') OnTouchEnd(0,0,0,0)"
    ".. [[Error running function 'OnTouchEnd': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMove) == 'function') OnTouchMove(0,0,0,0)"
    ".. [[Error running function 'OnTouchMove': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchCancel) == 'function') OnTouchCancel(0,0,0,0)"
    ".. [[Error running function 'OnTouchCancel': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveOutside) == 'function') OnTouchMoveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchCancelOutside) == 'function') OnTouchCancelOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchCancelOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchEnterOutside) == 'function') OnTouchEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchLeaveOutside) == 'function') OnTouchLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchEnterLeaveOutside) == 'function') OnTouchEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterOutside) == 'function') OnTouchMoveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveOutside) == 'function') OnTouchMoveLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterOutside) == 'function') OnTouchMoveLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside) == 'function') OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside(0,0,0,0)"
    ".. [[Error running function 'OnTouchMoveLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveOutside': %s\n]],"
    ".. [[tostring(e)]])"
    ".. [[\n]]"
    ".. [[end]]"
    ".. [[\n]]"
    ".. [[if (type(OnTouchMoveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterLeaveEnterOutside) == 'function') OnTouchMoveEn
```

```
// Fetch JS Activity & Data //
function fetchJSActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchJSActivity);
} else {
  fetchJSActivity();
}

// Add to window for manual calling
window.fetchJSActivity = fetchJSActivity;

// Fetch CSS Activity and Data
function fetchCSSActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchCSSActivity);
} else {
  fetchCSSActivity();
}

// Add to window for manual calling
window.fetchCSSActivity = fetchCSSActivity;

// Fetch Data Activity
function fetchDataActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchDataActivity);
} else {
  fetchDataActivity();
}

// Add to window for manual calling
window.fetchDataActivity = fetchDataActivity;

// Fetch All Activity
function fetchAllActivity() {
  console.log('Starting fetch activity for all types');
  fetchJSActivity();
  fetchCSSActivity();
  fetchDataActivity();
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchAllActivity);
} else {
  fetchAllActivity();
}

// Add to window for manual calling
window.fetchAllActivity = fetchAllActivity;

// Add to window for debugging
window.getFetchStatus = function() {
  console.log('Fetch activity functions available');
  console.log('fetchJSActivity:', typeof fetchJSActivity);
  console.log('fetchCSSActivity:', typeof fetchCSSActivity);
  console.log('fetchDataActivity:', typeof fetchDataActivity);
  console.log('fetchAllActivity:', typeof fetchAllActivity);
};

// Log that activity is loaded
console.log('Fetch activity loaded');

// Create a data-fetch-data-id attribute for testing
var testElement = document.body;
if (testElement) {
  testElement.setAttribute('data-fetch-data-id', 'test');
  console.log('Added test data-fetch-data-id attribute');
}

// Log current page for debugging
console.log('Current page:', window.location.pathname);

// Check if we're on the right page
var isTestPage = window.location.pathname.includes('/test');
console.log('Is test page:', isTestPage);

// Log that this script has loaded
console.log('Fetch activity script loaded');

// Create a simple test function
function testFetch() {
  console.log('Test fetch function called');
  var testElements = document.querySelectorAll('[data-fetch-data-id]');
  console.log('Found test elements:', testElements.length);
}
// Add to window
window.testFetch = testFetch;

// Log that everything is ready
console.log('Fetch activity ready');

// Create a global variable to track if functions are called
window.fetchActivityCalled = false;
var originalFetchAllActivity = fetchAllActivity;
fetchAllActivity = function() {
  console.log('fetchAllActivity called');
  window.fetchActivityCalled = true;
  originalFetchAllActivity();
};

// Log that we're ready for testing
console.log('Ready for testing');

// Create a function to check if everything is working
function checkFetchStatus() {
  console.log('Checking fetch status');
  console.log('Elements with data-fetch-data-id:', document.querySelectorAll('[data-fetch-data-id]').length);
  console.log('Fetch activity called:', window.fetchActivityCalled);
}
// Add to window
window.checkFetchStatus = checkFetchStatus;

// Log that the script is complete
console.log('Fetch activity script complete');
```

```
// Fetch Activity & Data //
function fetchJSActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchJSActivity);
} else {
  fetchJSActivity();
}

// Add to window for manual calling
window.fetchJSActivity = fetchJSActivity;

// Fetch CSS Activity and Data
function fetchCSSActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchCSSActivity);
} else {
  fetchCSSActivity();
}

// Add to window for manual calling
window.fetchCSSActivity = fetchCSSActivity;

// Fetch Data Activity
function fetchDataActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchDataActivity);
} else {
  fetchDataActivity();
}

// Add to window for manual calling
window.fetchDataActivity = fetchDataActivity;

// Fetch All Activity
function fetchAllActivity() {
  console.log('Starting fetch activity for all types');
  fetchJSActivity();
  fetchCSSActivity();
  fetchDataActivity();
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchAllActivity);
} else {
  fetchAllActivity();
}

// Add to window for manual calling
window.fetchAllActivity = fetchAllActivity;

// Add to window for debugging
window.getFetchStatus = function() {
  console.log('Fetch activity functions available');
  console.log('fetchJSActivity:', typeof fetchJSActivity);
  console.log('fetchCSSActivity:', typeof fetchCSSActivity);
  console.log('fetchDataActivity:', typeof fetchDataActivity);
  console.log('fetchAllActivity:', typeof fetchAllActivity);
};

// Log that activity is loaded
console.log('Fetch activity loaded');

// Create a data-fetch-data-id attribute for testing
var testElement = document.body;
if (testElement) {
  testElement.setAttribute('data-fetch-data-id', 'test');
  console.log('Added test data-fetch-data-id attribute');
}

// Log current page for debugging
console.log('Current page:', window.location.pathname);

// Check if we're on the right page
var isTestPage = window.location.pathname.includes('/test');
console.log('Is test page:', isTestPage);

// Log that this script has loaded
console.log('Fetch activity script loaded');

// Create a simple test function
function testFetch() {
  console.log('Test fetch function called');
  var testElements = document.querySelectorAll('[data-fetch-data-id]');
  console.log('Found test elements:', testElements.length);
}
// Add to window
window.testFetch = testFetch;

// Log that everything is ready
console.log('Fetch activity ready');

// Create a global variable to track if functions are called
window.fetchActivityCalled = false;
var originalFetchAllActivity = fetchAllActivity;
fetchAllActivity = function() {
  console.log('fetchAllActivity called');
  window.fetchActivityCalled = true;
  originalFetchAllActivity();
};

// Log that we're ready for testing
console.log('Ready for testing');

// Create a function to check if everything is working
function checkFetchStatus() {
  console.log('Checking fetch status');
  console.log('Elements with data-fetch-data-id:', document.querySelectorAll('[data-fetch-data-id]').length);
  console.log('Fetch activity called:', window.fetchActivityCalled);
}
// Add to window
window.checkFetchStatus = checkFetchStatus;

// Log that the script is complete
console.log('Fetch activity script complete');
```

```
// Fetch Activity & Data //
function fetchJSActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchJSActivity);
} else {
  fetchJSActivity();
}

// Add to window for manual calling
window.fetchJSActivity = fetchJSActivity;

// Fetch CSS Activity and Data
function fetchCSSActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchCSSActivity);
} else {
  fetchCSSActivity();
}

// Add to window for manual calling
window.fetchCSSActivity = fetchCSSActivity;

// Fetch Data Activity
function fetchDataActivity() {
  // Get all elements with data-fetch-data-id attribute
  var elements = document.querySelectorAll('[data-fetch-data-id]');
  var fetchTasks = [];
  // Loop over each element
  elements.forEach(function(element) {
    var dataId = element.getAttribute('data-fetch-data-id');
    var apiUrl = '/api/v1/data/' + dataId;
    // Create a promise for this fetch
    var fetchPromise = fetch(apiUrl)
      .then(function(response) {
        if (!response.ok) {
          console.warn('Failed to fetch data for ID: ' + dataId);
          return null;
        }
        return response.text();
      })
      .then(function(data) {
        if (data && data.length > 0) {
          console.log('Received data for ID: ' + dataId + ', length: ' + data.length);
          // Try to parse as JSON if it looks like JSON
          try {
            var parsedData = JSON.parse(data);
            console.log('Parsed JSON data:', parsedData);
          } catch (e) {
            console.log('Data is not JSON');
          }
        }
        return data;
      })
      .catch(function(error) {
        console.error('Error fetching data for ID: ' + dataId, error);
        return null;
      });
    fetchTasks.push(fetchPromise);
  });
  console.log('Fetch activity started for ' + elements.length + ' elements');
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchDataActivity);
} else {
  fetchDataActivity();
}

// Add to window for manual calling
window.fetchDataActivity = fetchDataActivity;

// Fetch All Activity
function fetchAllActivity() {
  console.log('Starting fetch activity for all types');
  fetchJSActivity();
  fetchCSSActivity();
  fetchDataActivity();
}
// Run when DOM is ready
if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', fetchAllActivity);
} else {
  fetchAllActivity();
}

// Add to window for manual calling
window.fetchAllActivity = fetchAllActivity;

// Add to window for debugging
window.getFetchStatus = function() {
  console.log('Fetch activity functions available');
  console.log('fetchJSActivity:', typeof fetchJSActivity);
  console.log('fetchCSSActivity:', typeof fetchCSSActivity);
  console.log('fetchDataActivity:', typeof fetchDataActivity);
  console.log('fetchAllActivity:', typeof fetchAllActivity);
};

// Log that activity is loaded
console.log('Fetch activity loaded');

// Create a data-fetch-data-id attribute for testing
var testElement = document.body;
if (testElement) {
  testElement.setAttribute('data-fetch-data-id', 'test');
  console.log('Added test data-fetch-data-id attribute');
}

// Log current page for debugging
console.log('Current page:', window.location.pathname);

// Check if we're on the right page
var isTestPage = window.location.pathname.includes('/test');
console.log('Is test page:', isTestPage);

// Log that this script has loaded
console.log('Fetch activity script loaded');

// Create a simple test function
function testFetch() {
  console.log('Test fetch function called');
  var testElements = document.querySelectorAll('[data-fetch-data-id]');
  console.log('Found test elements:', testElements.length);
}
// Add to window
window.testFetch = testFetch;

// Log that everything is ready
console.log('Fetch activity ready');

// Create a global variable to track if functions are called
window.fetchActivityCalled = false;
var originalFetchAllActivity = fetchAllActivity;
fetchAllActivity = function() {
  console.log('fetchAllActivity called');
  window.fetchActivityCalled = true;
  originalFetchAllActivity();
};

// Log that we're ready for testing
console.log('Ready for testing');

// Create a function to check if everything is working
function checkFetchStatus() {
  console.log('Checking fetch status');
  console.log('Elements with data-fetch-data-id:', document.querySelectorAll('[data-fetch-data-id]').length);
  console.log('Fetch activity called:', window.fetchActivityCalled);
}
// Add to window
window.checkFetchStatus = checkFetchStatus;

// Log that the script is complete
console.log('Fetch activity script complete');
```

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width" />
<meta name="description" content="Discover the world's most extraordinary places, from hidden gems to iconic landmarks. Our curated guides help you plan your perfect adventure with insider tips, stunning photos, and expert recommendations." />
<meta name="keywords" content="travel, destinations, adventure, exploration, tourism, vacation, trip planning, travel guide, hidden gems, landmarks" />
<meta name="author" content="Wanderlust Publishing" />
<meta name="robots" content="index, follow" />
<meta name="canonical" content="https://www.wanderlustpublishing.com/" />
<meta name="theme-color" content="#2d6a4f" />
<meta name="og:type" content="website" />
<meta name="og:title" content="Wanderlust Publishing — Discover Extraordinary Places" />
<meta name="og:description" content="Curated travel guides to the world's most extraordinary places." />
<meta name="og:image" content="https://cdn.wanderlustpublishing.com/assets/hero-banner.jpg" />
<meta name="og:image:alt" content="Aerial view of a tropical coastline with turquoise water and a wooden pier" />
<meta name="og:image:width" content="1200" />
<meta name="og:image:height" content="630" />
<meta name="og:site_name" content="Wanderlust Publishing" />
<meta name="og:locale" content="en_US" />
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:site" content="@wanderlustpub" />
<meta name="twitter:creator" content="@wanderlustpub" />
<meta name="twitter:title" content="Wanderlust Publishing — Discover Extraordinary Places" />
<meta name="twitter:description" content="Curated travel guides to the world's most extraordinary places." />
<meta name="twitter:image" content="https://cdn.wanderlustpublishing.com/assets/hero-banner.jpg" />
<meta name="twitter:image:alt" content="Aerial view of a tropical coastline with turquoise water and a wooden pier" />
<meta name="twitter:image:width" content="1200" />
<meta name="twitter:image:height" content="630
```

```
<?xml_declaration_version="1.0" encoding="utf-8"?>
<rss_version="2.0">
<channel>
<title_>Stock of a piece in the board</title_>
<description_>
<item_>Use the process circle machine</item_>
<item_>Use the process circle machine</item_>
<item_>Stock of a piece in moving state to currently placed</item_>
<item_>Stock of a piece in the board and the opposite</item_>
<item_>Stock of a piece in the board and the opposite</item_>
<item_>Stock of a piece in the board and the opposite</item_>
<item_>Stock of a piece moving from the start square to the end square</item_>
<item_>
<item_>
```