Common.js:مدیاویکی

نکته: پس از انتشار ممکن است برای دیدن تغییرات نیاز باشد که حافظهٔ نهانی مرورگر خود را پاک کنید.

- **فایرفاکس / سافاری:** کلید *Shift* را نگه دارید و روی دکمهٔ *Reload* کلیک کنید، یا کلیدهای *Ctrl-F5* یا *Ctrl-R* را با هم فشار دهید (در رایانه‌های اپل مکینتاش کلیدهای ⌘-*R*)
- **گوگل کروم:** کلیدهای *Ctrl+Shift+R* را با هم فشار دهید (در رایانه‌های اپل مکینتاش کلیدهای ⌘-*Shift*-*R*)
- **Edge:** کلید *Ctrl* را نگه‌دارید و روی دکمهٔ *Refresh* کلیک کنید، یا کلیدهای *Ctrl-F5* را با هم فشار دهید

```
const fadeElements = document.querySelectorAll('.fadein');

const observer = new IntersectionObserver((entries) => {
  entries.forEach(entry => {
    // وقتی حداقل ۶۰٪ المنت داخل ویوپورت است
    if (entry.isIntersecting && entry.intersectionRatio >= 0.6) {
      entry.target.classList.remove('fadein');
      entry.target.classList.add('fadedin');
      observer.unobserve(entry.target); // فقط یک بار
    }
  });
}, {
  root: null,         // ویوپورت
  threshold: 0.6,     // ۶۰٪
  rootMargin: '0px'   // بدون حاشیه
});

fadeElements.forEach(el => observer.observe(el));

/* counter effect */

const counters = document.querySelectorAll('.value');
const speed = 200;

counters.forEach( counter => {
   const animate = () => {
      const value = +counter.getAttribute('akhi');
      const data = +counter.innerText;
      const time = value / speed;
     if(data < value) {
          counter.innerText = Math.ceil(data + time);
          setTimeout(animate, 1);
        }else{
          counter.innerText = value;
        }
   }
```

```
   animate();
});

/* adding links to the header */
(function() {
   document.querySelector('.citizen-
header__start').insertAdjacentHTML('beforeend', '<span style="margin-right:
20px">برگریزون</span><span style="margin-right: 20px"><a
href="/view/فهرست الفبایی">فهرست_الفبایی:ویکی_معرفت</a></span><span
style="margin-right: 20px"><a hr ef="">کتابت رو بسازکتابت رو بسازکتابت رو بسازکتابت رو بساز</a></span><span
style="margin-right: 20px">درباره ما</span>');
})();
(function() {
   document.querySelector('.citizen-
header__start').insertAdjacentHTML('beforeend', '<div
class="socialmedia"><img src="/images/e/ef/Eitaa.png" style="margin-right:
10px"></img><img src="/images/b/b7/Instagram.png" style="margin-right:
10px"></img><img src="/images/a/af/Youtube.png" style="margin-right:
10px"></img></div>');
})();
/* global mw, $ */
// Editing-related javascripts
if (mw.config.get("wgAction") === "edit" || mw.config.get("wgAction") ===
"submit" || mw.config.get("wgCanonicalSpecialPageName") === "Upload") {
  /* scripts specific to editing pages */
  importScript("مدیاویکی:Common.js/edit.js");
}
/** Magic editintros ***************************************************
 *
 *  Description: Adds editintros on disambiguation pages and BLP pages.
 *  Maintainers: [[User:RockMFR]]
 */
function addEditIntro(name) {
  $(".mw-editsection, #ca-edit").find("a").each(function(i, el) {
    el.href = $(this).attr("href") + "&editintro=" + name;
  });
}

if (mw.config.get("wgNamespaceNumber") === 0) {
  $(function() {
    if (document.getElementById("disambigbox")) {
      addEditIntro("Template:Disambig_editintro");
    }
  });

  $(function() {
    var cats = mw.config.get("wgCategories");
    if (!cats) {
      return;
    }
    if ($.inArray("افراد زنده", cats) !== -1) {
```

```
      addEditIntro("Template:BLP_editintro");
    }
  });
}

/**
 * @source https://www.mediawiki.org/wiki/Snippets/Load_JS_and_CSS_by_URL
 * @revision 2020-04-04
 */
mw.loader.using( ['mediawiki.util'], function () {
        var extraCSS = mw.util.getParamValue( 'withCSS' ),
                extraJS = mw.util.getParamValue( 'withJS' );

        if ( extraCSS ) {
                // WARNING: DO NOT REMOVE THIS "IF" - REQUIRED FOR SECURITY
(against XSS/CSRF attacks)
                if ( /^MediaWiki:[^&<>=%#]*\.css$/.test( extraCSS ) ) {
                        mw.loader.load( '/w/index.php?title=' +
encodeURIComponent( extraCSS ) + '&action=raw&ctype=text/css', 'text/css' );
                } else {
                        mw.notify( 'Only pages from the MediaWiki namespace
are allowed.', { title: 'Invalid withCSS value' } );
                }
        }

        if ( extraJS ) {
                // WARNING: DO NOT REMOVE THIS "IF" - REQUIRED FOR SECURITY
(against XSS/CSRF attacks)
                if ( /^MediaWiki:[^&<>=%#]*\.js$/.test( extraJS ) ) {
                        mw.loader.load( '/w/index.php?title=' +
encodeURIComponent( extraJS ) + '&action=raw&ctype=text/javascript' );
                } else {
                        mw.notify( 'Only pages from the MediaWiki namespace
are allowed.', { title: 'Invalid withJS value' } );
                }
        }
});

/**
 * Opening preload.js to add interactive training keys
 */
mw.loader.using(["mediawiki.util"], function() {
  if (mw.util.getParamValue("preload")) {
    importScript("MediaWiki:Preloaded.js");
  }
});

/**
 * Collapsible tables; reimplemented with mw-collapsible
 * Styling is also in place to avoid FOUC
 *
```

```
 * Allows tables to be collapsed, showing only the header. See
[[Help:Collapsing]].
 * @version 3.0.0 (2018-05-20)
 * @source
https://www.mediawiki.org/wiki/MediaWiki:Gadget-collapsibleTables.js
 * @author [[User:R. Koot]]
 * @author [[User:Krinkle]]
 * @author [[User:TheDJ]]
 * @deprecated Since MediaWiki 1.20: Use class="mw-collapsible" instead which
 * is supported in MediaWiki core. Shimmable since MediaWiki 1.32
 *
 * @param {jQuery} $content
 */
function makeCollapsibleMwCollapsible( $content ) {
        var $tables = $content
                .find( 'table.collapsible:not(.mw-collapsible)' )
                .addClass( 'mw-collapsible' );

        $.each( $tables, function ( index, table ) {
                // mw.log.warn( 'This page is using the deprecated class
collapsible. Please replace it with mw-collapsible.');
                if ( $( table ).hasClass( 'collapsed' ) ) {
                        $( table ).addClass( 'mw-collapsed' );
                        // mw.log.warn( 'This page is using the deprecated
class collapsed. Please replace it with mw-collapsed.');
                }
        } );
        if ( $tables.length > 0 ) {
                mw.loader.using( 'jquery.makeCollapsible' ).then( function ()
{
                        $tables.makeCollapsible();
                } );
        }
}
mw.hook( 'wikipage.content' ).add( makeCollapsibleMwCollapsible );

/**
 * Add support to mw-collapsible for autocollapse, innercollapse and
outercollapse
 *
 * Maintainers: TheDJ
 */
function mwCollapsibleSetup( $collapsibleContent ) {
        var $element,
                $toggle,
                autoCollapseThreshold = 2;
        $.each( $collapsibleContent, function ( index, element ) {
                $element = $( element );
                if ( $element.hasClass( 'collapsible' ) ) {
                        $element.find( 'tr:first > th:first' ).prepend(
$element.find( 'tr:first > * > .mw-collapsible-toggle' ) );
```

```
                }
                if ( $collapsibleContent.length >= autoCollapseThreshold &&
$element.hasClass( 'autocollapse' ) ) {
                        $element.data( 'mw-collapsible' ).collapse();
                } else if ( $element.hasClass( 'innercollapse' ) ) {
                        if ( $element.parents( '.outercollapse' ).length > 0
) {
                                $element.data( 'mw-collapsible' ).collapse();
                        }
                }
                // because of colored backgrounds, style the link in the text
color
                // to ensure accessible contrast
                $toggle = $element.find( '.mw-collapsible-toggle' );
                if ( $toggle.length ) {
                        // Make the toggle inherit text color
                        if ( $toggle.parent()[ 0 ].style.color ) {
                                $toggle.find( 'a' ).css( 'color', 'inherit'
);
                        }
                }
        } );
}

mw.hook( 'wikipage.collapsibleContent' ).add( mwCollapsibleSetup );

/**
 * Dynamic Navigation Bars (experimental)
 *
 * Description: See [[Wikipedia:NavFrame]].
 * Maintainers: UNMAINTAINED
 */

var collapseCaption = 'بسته';
var expandCaption = 'باز';

// Set up the words in your language
var navigationBarHide = '[' + collapseCaption + ']';
var navigationBarShow = '[' + expandCaption + ']';

/**
 * Shows and hides content and picture (if available) of navigation bars.
 *
 * @param {number} indexNavigationBar The index of navigation bar to be
toggled
 * @param {jQuery.Event} event Event object
 * @return {boolean}
 */
function toggleNavigationBar( indexNavigationBar, event ) {
        var navToggle = document.getElementById( 'NavToggle' +
indexNavigationBar );
```

```
        var navFrame = document.getElementById( 'NavFrame' +
indexNavigationBar );
        var navChild;

        if ( !navFrame || !navToggle ) {
                return false;
        }

        // If shown now
        if ( navToggle.firstChild.data === navigationBarHide ) {
                for ( navChild = navFrame.firstChild; navChild !== null;
navChild = navChild.nextSibling ) {
                        if ( $( navChild ).hasClass( 'NavContent' ) ) {
                                navChild.style.display = 'none';
                        }
                }
                navToggle.firstChild.data = navigationBarShow;

        // If hidden now
        } else if ( navToggle.firstChild.data === navigationBarShow ) {
                for ( navChild = navFrame.firstChild; navChild !== null;
navChild = navChild.nextSibling ) {
                        if ( $( navChild ).hasClass( 'NavContent' ) ) {
                                navChild.style.display = 'block';
                        }
                }
                navToggle.firstChild.data = navigationBarHide;
        }

        event.preventDefault();
}

/**
 * Adds show/hide-button to navigation bars.
 *
 * @param {jQuery} $content
 */
function createNavigationBarToggleButton( $content ) {
        var j, navChild, navToggle, navToggleText, isCollapsed,
                indexNavigationBar = 0;
        // Iterate over all < div >-elements
        var $divs = $content.find( 'div.NavFrame:not(.mw-collapsible)' );
        $divs.each( function ( i, navFrame ) {
                indexNavigationBar++;
                navToggle = document.createElement( 'a' );
                navToggle.className = 'NavToggle';
                navToggle.setAttribute( 'id', 'NavToggle' +
indexNavigationBar );
                navToggle.setAttribute( 'href', '#' );
                $( navToggle ).on( 'click', $.proxy( toggleNavigationBar,
null, indexNavigationBar ) );
```

```
                isCollapsed = $( navFrame ).hasClass( 'collapsed' );
                /**
                 * Check if any children are already hidden.  This loop is
here for backwards compatibility:
                 * the old way of making NavFrames start out collapsed was to
manually add style="display:none"
                 * to all the NavPic/NavContent elements.  Since this was bad
for accessibility (no way to make
                 * the content visible without JavaScript support), the new
recommended way is to add the class
                 * "collapsed" to the NavFrame itself, just like with
collapsible tables.
                 */
                for ( navChild = navFrame.firstChild; navChild !== null &&
!isCollapsed; navChild = navChild.nextSibling ) {
                        if ( $( navChild ).hasClass( 'NavPic' ) || $(
navChild ).hasClass( 'NavContent' ) ) {
                                if ( navChild.style.display === 'none' ) {
                                        isCollapsed = true;
                                }
                        }
                }
                if ( isCollapsed ) {
                        for ( navChild = navFrame.firstChild; navChild !==
null; navChild = navChild.nextSibling ) {
                                if ( $( navChild ).hasClass( 'NavPic' ) || $(
navChild ).hasClass( 'NavContent' ) ) {
                                        navChild.style.display = 'none';
                                }
                        }
                }
                navToggleText = document.createTextNode( isCollapsed ?
navigationBarShow : navigationBarHide );
                navToggle.appendChild( navToggleText );

                // Find the NavHead and attach the toggle link (Must be this
complicated because Moz's firstChild handling is borked)
                for ( j = 0; j < navFrame.childNodes.length; j++ ) {
                        if ( $( navFrame.childNodes[ j ] ).hasClass(
'NavHead' ) ) {
                                navToggle.style.color = navFrame.childNodes[
j ].style.color;
                                navFrame.childNodes[ j ].appendChild(
navToggle );
                        }
                }
                navFrame.setAttribute( 'id', 'NavFrame' + indexNavigationBar
);
        } );
}
```

```
mw.hook( 'wikipage.content' ).add( createNavigationBarToggleButton );

/** Dynamic Navigation Bars (experimental)
************************************
 *
 *  Description: See [[Wikipedia:NavFrame]].
 *  Maintainers: UNMAINTAINED
 */

// set up the words in your language
var NavigationBarHide = "[" + collapseCaption + "]";
var NavigationBarShow = "[" + expandCaption + "]";

// shows and hides content and picture (if available) of navigation bars
// Parameters:
//     indexNavigationBar: the index of navigation bar to be toggled
window.toggleNavigationBar = function(indexNavigationBar, event) {
  var NavToggle = document.getElementById("NavToggle" + indexNavigationBar);
  var NavFrame = document.getElementById("NavFrame" + indexNavigationBar);
  var NavChild;

  if (!NavFrame || !NavToggle) {
    return false;
  }

  /* if shown now */
  if (NavToggle.firstChild.data === NavigationBarHide) {
    for (NavChild = NavFrame.firstChild; NavChild != null; NavChild =
NavChild.nextSibling) {
      if ($(NavChild).hasClass("NavContent") ||
$(NavChild).hasClass("NavPic")) {
        NavChild.style.display = "none";
      }
    }
    NavToggle.firstChild.data = NavigationBarShow;

    /* if hidden now */
  } else if (NavToggle.firstChild.data === NavigationBarShow) {
    for (NavChild = NavFrame.firstChild; NavChild != null; NavChild =
NavChild.nextSibling) {
      if ($(NavChild).hasClass("NavContent") ||
$(NavChild).hasClass("NavPic")) {
        NavChild.style.display = "block";
      }
    }
    NavToggle.firstChild.data = NavigationBarHide;
  }

  event.preventDefault();
};
```

```
/* adds show/hide-button to navigation bars */
function createNavigationBarToggleButton() {
  var indexNavigationBar = 0;
  var NavFrame;
  var NavChild;
  /* iterate over all < div >-elements */
  var divs = document.getElementsByTagName("div");
  for (var i = 0;
    (NavFrame = divs[i]); i++) {
    /* if found a navigation bar */
    if ($(NavFrame).hasClass("NavFrame")) {

      indexNavigationBar++;
      var NavToggle = document.createElement("a");
      NavToggle.className = "NavToggle";
      NavToggle.setAttribute("id", "NavToggle" + indexNavigationBar);
      NavToggle.setAttribute("href", "#");
      $(NavToggle).on("click", $.proxy(window.toggleNavigationBar, window,
indexNavigationBar));

      var isCollapsed = $(NavFrame).hasClass("collapsed");
      /**
       * Check if any children are already hidden.  This loop is here for
backwards compatibility:
       * the old way of making NavFrames start out collapsed was to manually
add style="display:none"
       * to all the NavPic/NavContent elements.  Since this was bad for
accessibility (no way to make
       * the content visible without JavaScript support), the new recommended
way is to add the class
       * "collapsed" to the NavFrame itself, just like with collapsible
tables.
       */
      for (NavChild = NavFrame.firstChild; NavChild != null && !isCollapsed;
NavChild = NavChild.nextSibling) {
        if ($(NavChild).hasClass("NavPic") ||
$(NavChild).hasClass("NavContent")) {
          if (NavChild.style.display === "none") {
            isCollapsed = true;
          }
        }
      }
      if (isCollapsed) {
        for (NavChild = NavFrame.firstChild; NavChild != null; NavChild =
NavChild.nextSibling) {
          if ($(NavChild).hasClass("NavPic") ||
$(NavChild).hasClass("NavContent")) {
            NavChild.style.display = "none";
          }
        }
      }
```

```
      var NavToggleText = document.createTextNode(isCollapsed ?
NavigationBarShow : NavigationBarHide);
      NavToggle.appendChild(NavToggleText);

      /* Find the NavHead and attach the toggle link (Must be this
complicated because Moz"s firstChild handling is borked) */
      for (var j = 0; j < NavFrame.childNodes.length; j++) {
        if ($(NavFrame.childNodes[j]).hasClass("NavHead")) {
          NavToggle.style.color = NavFrame.childNodes[j].style.color;
          NavFrame.childNodes[j].appendChild(NavToggle);
        }
      }
      NavFrame.setAttribute("id", "NavFrame" + indexNavigationBar);
    }
  }
}

// $(createNavigationBarToggleButton);

/** Main Page layout fixes
*************************************************************
 *
 *  Description: Adds an additional link to the complete list of languages
available.
 *  Maintainers: [[User:AzaToth]], [[User:R. Koot]], [[User:Alex Smotrov]]
 */
mw.loader.using(["mediawiki.util"], function() {
  if (["بحث:صفحه_اصلی" ,"صفحهٔ_اصلی"].indexOf(mw.config.get("wgPageName"))
!== -1) {
    $(function() {
      mw.util.addPortletLink("p-lang",
"//meta.wikimedia.org/wiki/List_of_Wikipedias/fa", "فهرست کامل", "interwiki-
completelist", "فهرست کامل");
    });
  }
});

/**
 * WikiMiniAtlas
 *
 * Description: WikiMiniAtlas is a popup click and drag world map.
 *              This script causes all of our coordinate links to display the
WikiMiniAtlas popup button.
 *              The script itself is located on meta because it is used by
many projects.
 *              See [[Meta:WikiMiniAtlas]] for more information.
 * Maintainers: [[User:Dschwen]]
 */
(function() {
  var require_wikiminiatlas = false;
  var coord_filter = /geohack/;
```

```
  $(function() {
    $("a.external.text").each(function(key, link) {
      if (link.href && coord_filter.exec(link.href)) {
        require_wikiminiatlas = true;
        // break from loop
        return false;
      }
    });
    if ($("div.kmldata").length) {
      require_wikiminiatlas = true;
    }
    if (require_wikiminiatlas) {
mw.loader.load("//meta.wikimedia.org/w/index.php?title=MediaWiki:Wikiminiatla
s.js&action=raw&ctype=text/javascript");
    }
  });
})();

/**
 * Source codes pages direction and and pre white-space
 */
mw.loader.using(["mediawiki.util"], function() {
  if ([2, 4, 8].indexOf(mw.config.get("wgNamespaceNumber")) !== -1) {
    if (/\.(js|css|py)$/.test(mw.config.get("wgPageName"))) {
      mw.util.addCSS(".diff-addedline, .diff-deletedline, .diff-context {
direction: ltr; } " +
        "#wpTextbox1 { direction: ltr; }");
    }
  }

  // Newbies are misusing wikilove instead using talk pages add new section
  // hiding WikiLove from not autoconfirmed users or without more than 100
edits
  if ($.inArray("autoconfirmed", mw.config.get("wgUserGroups")) === -1 ||
mw.config.get("wgUserEditCount") < 100) {
    mw.util.addCSS("#ca-wikilove { display: none; }");
  }
});
/**
 * قراردادن فرمول‌های ریاضی در سمت چپ مناسب برای ویکی‌های راست به چپ
 */
$(function() {
  $("span.mwe-math-mathml-inline").each(function() {
    if ($(this).parent().parent().text().trim() ===
$(this).parent().text().trim() && $(this).parent().parent().is("p, dd")) {
      $(this).parent().parent().css({
        direction: "ltr",
        padding: "1em 0"
      });
    }
  });
```

```
});

// هشدار به کاربرها در مورد عنوان ایمیل
$(function() {
  if ( $('#mw-input-wpSubject').parents('tr').length > 0 ) {
    $('#mw-input-wpSubject').parents('tr').before('<tr><td></td><td
style="color: red; font-weight: bold">لطفاً عبارت Wikipedia e-mail را با یک
عنوان مناسب جایگزین کنید!</td><tr>');
  }
});

// Make refrence number that you clicked on bold for proper navigation
// Author: User:Ladsgroup
$( 'sup.reference' ).click( function () {
    var citeId = $( this ).attr( 'id' );
    $( 'sup > a[href="#' + citeId + '"]').css('color', '#2a4b8d').css('font-
weight', 'bold');
    $( 'sup > a[href="#' + citeId + '"]').click( function() {
        $( this ).css('color', '').css('font-weight', '');
    } );
} );

// Hide the FileExporter portlet link if the file is not free content
$(function() {
  if ($('#ca-fileExporter').length == 1 &
$(mw.config.get('wgCategories')).filter(['محتویات غیر آزاد', 'تصویرهای با
منبع نامعلوم', 'تصویرهای با پدیدآور نامشخص']).length > 0) {
        $('#ca-fileExporter').hide();
  }
});

/* اضافه کردن تیتر برای یادداشت ها */
$( function() {
    if($('#reflist-notes .references').text().length>10) {
        $('#reflist-notes').before('<h2>یادداشت‌ها</h2>');
    }
});
```