-DOM (כך מכונה דף אינטרנט ב-javascript)

הקוד הבסיסי של JS:

<html>

<head>

  <meta charset="utf-8">

</head>

<body>

**<script>**

alert("שלום עולם");//דוגמא

**</script>**

</body>

</html>

הגדרת משתנה והדפסה שלו על המסך (תצוגה)

**var** animal = "פרה";

document.write(animal);

זה מדפיס מימין לשמאל, ברצף ואפשר כמובן להגדיר סוף שורה וכו'.

אם נגדיר מספר (אפילו שבר) לא נשים במרכאות (שמוצב לתוך משתנה).

כתיבת מרכאות (בתוך מחרוזת) כותבים כך /"

חיבור מחרוזות:

alert("יש לי " + myAnimal);

פונקציות

**function** hello(name) {

    alert("שלום " + name);

}

אם יש קריאה כזו:

hello("תקווה","ברוך");

אז הפרמטר הראשון שהועבר הוא "תקווה" כרגיל כמו באנגלית.

פרמטרים ברירת מחדל:

**function** hello(from, to) {

if(from === undefined){

from = "א";

}

או בצורה מקוצרת:

from = from || "א";

אפשר להחזיר רגיל ערך מפונקציה ע"י ואין צורך להצהיר בפונקציה שמחזירה משהו כי זו שפת סקריפט.

הגדרת פונקציה אנונימית שהיא פונקציה ללא שם ושימושית עבור אירועים למשל:

function (){}

כל ה-if , הלולאות, עובד כרגיל כמו בסינטקס של שפת C.

**הדרך הנכונה לשילוב קוד JavaScript באתר**

**קיימות 2 דרכים עיקריות להוסיף קוד JavaScript לאתר:**

1. קוד שנמצא בתוך הדף עצמו, כפי שראינו במדריכים הקודמים.
2. לשים את הקוד בקובץ JavaScript נפרד.

צריך לשים לב שאת הסקריפטים של JavaScript מעדיפים לשים בתחתית ה-body, מתחת לכל ההטמ"ל, וקצת מעל לתגית ה-body הסוגרת. הסיבה לכך, היא שהטעינה של קוד JavaScript מאטה את הדף, ולכן נעדיף לטעון את ה-JavaScript אחרון, כדי שלא יפריע ליתר הדף להטען.

השיטה של הוספת קוד ה-JavaScript לכל דף בנפרד אינה מומלצת משתי סיבות:

* לא ניתן לשמור בקאש של הדפדפן דבר שמאט את טעינת הדף בביקורים חוזרים.
* צריך לכתוב את אותו קוד בדפים שונים, שזה סתם לא חכם.

לכן, עדיף לטעון את ה-JavaScript מקובץ js חיצוני.

טעינה של קובץ JavaScript חיצוני

כדי לטעון את קובץ ה-JavaScript החיצוני אנחנו צריכים שיהיו לנו שני קבצים:

קובץ הטמ"ל עם סיומת html או קובץ PHP שמייצר קובץ תבנית. לדוגמה, index.html

**קובץ JavaScript עם סיומת js. לדוגמה, script.js**

כדי להשתמש בקובץ ה-JavaScript החיצוני, בקובץ ההטמ"ל, **נוסיף לתגית ה-script את השדה src הכולל את מקום הקובץ**. כך נראית תגית ה-script, אם אנחנו רוצים לטעון את הקובץ script.js, מתוך התיקייה scripts:

<script src="scripts/script.js"></script>

בנוסף, הדפדפן שומר את קבצי ה-JavaScript בקאש, וזה גורם להאצת מהירות הגלישה כשהמשתמשים חוזרים לבקר באתר.

מערכים די דומה מלבד שאין סוג משתנה וגם מבצע הכרזה ואתחול:

**var** numbers = [42, 13, 7, 6, 1];//כלומר אין סוגריים מסולסלות ואין הכרזת גודל

אפשר גם ככה להציב ערכים למערך:

var a = [];

a[2]=4;

a[3]=1;

פריט שלא קיים (למשל במערך) ערכו הוא undefined (כלומר סוג של Null) ככה בודקים זאת:

הערה- undefined אפשר לרשום גם בלי ' ' לפי מה שבדקתי.

הוספת פריט נוסף לסוף מערך קיים:

vip.push("1.75");

החזרת האיבר האחרון במערך כולל הסרה שלו מהמערך:

animals.pop();

הפיכת סדר איברים במערך:

animals.**reverse**();

מחיקת מספר פריטים מהמערך (אחרי זה גודלו משתנה ומתקצר):

vip.splice(0,2);

0 אומר למחוק החל מאינדקס 0 (כולל) ולמחזוק 2 איברים (כלומר אינדקס 0 ו-1).

Vip.length//יחזיר מספר הפריטים במערך

**var** animals = farmAnimals.**concat**(wildAnimals);//חיבור מערך לסוף המערך הראשון

**var** stringOfAnimals = animals.**join**();//הופך מערך למחרוזת שלמה מופרדת בפסיקים

join(" # ");//אפשר לרשום תו אחר ואז יפריד באמצעות סולמית למשל

מערכים אסוציאטיבי:

**var** vip = [];

vip["first\_name"] = "א";

vip["last\_name"] = "ב";

vip["girlfriend"] = "ד";

vip["x\_girlfriend"] = "ח";

שליפת ערך שמתאים למפתח:

**var** x = vip["first\_name"];

document.write(x);// א

אירועים (מאזינים) ב-JS:

document.getElementById(**"**myBtn**"**).**addEventListener**("**click**", myFunction);

החלק האדום = קבלת סוג האלמנט שעבורו קרה האירוע, למשל כפתור myBtn.

החלק הכחול=הוספת המאזין

החלק הירוק=סוג האירוע (במקרה הזה עבור קליל בעכבר) כאשר הפונקציה המאזינה והיא הפונקציה הביצועית בעת לחיצה myFunction

הערה- לאותו אלמנט (למשל כפתור) אפשר להוסיף מסמפר מאזינים שונים, למשל קליק, וגם מעבר עם העכבר מעל הכפתור וכו'.

אפשר לממש פונקציה אנונימית בעת הוספת המאזין בצורה כזו:

**addEventListener**("**click**", function(){ **popUp();** });

שימו לב, שקוראים לפונקציה popUp בתוך פונקציה אנונימית כדי למנוע קריאה לפונקציה עם טעינת הדף. כי כאשר הדף נטען זה כביכול אוטומטית מבצע קליק בכפתור מה שכמובן לא נכון, לכן נעטוף בפונקציה אנונימית את הקריאה.

רשימת אירועים ב-JS:

|  |  |
| --- | --- |
| **האירוע** | **מתי הוא יורה?** |
| **click** | כשמקליקים על אלמנט |
| **mouseover / mouseout** | כשנכנסים או יוצאים עם העכבר מאלמנט |
| **keyup / keydown** | כשלוחצים על כפתור במקלדת |
| **focus / blur** | כשנכנסים או יוצאים משדה בטופס |
| **select** | כשבוחרים ערך מ- select list |
| **submit** | כשמגישים טופס |
| **load** | כשהדף סיים להטען |
| **unload** | כשיוצאים מדף |

**הדרך היעילה למניעת אירועים דיפולטיים**

נרצה שלחיצה על כפתור הטופס לא תשלח אותו ישירות בפוסט לשרת, אלא שקודם תתבצע עליו ולידציה באמצעות פונקציות של javascript.

javascript מספיק חכם גם כדי למנוע מאורעות דיפולטיים, והוא עושה זאת באמצעות העברה לפונקציה של פרמטר המאורע, e, וביטול המאורע באמצעות preventDefault.

function ( e ) {

e.preventDefault();

}

אנו מעבירים לפונקציה את הפרמטר e (קיצור של event), ומשתמשים בפרמטר כדי למנוע את האירוע הדיפולטי באמצעות : e.preventDefault().

דוגמת הקוד הבאה, מאזינה ללחיצה על קישור, אבל מונעת את המעבר לדף אחר (מה שקורה בד"כ כברירת מחדל בעת לחיצה על קישור), ומקפיצה פופ-אפ במקום.

…

<a href="" id="myLink">

לחץ עליי!

</a>

document.getElementById("myLink").

**addEventListener**("**click**", function( **e** ){ **e.preventDefault();** popUp();});

חשוב – שרושמים קוד JS חשוב שהקוד יהיה רשום (למשל) מתחת לאלמנט.

במקרה הזה מתחת ל a href כי אם יהיה רשום גם מעל ה-body זה ייבדק רק בהתחלה.

אובייקטים ב-JS

כך מגדירים מחלקה (ע"י פונקצית בנאי):

function Auto(type, fuel) {

this.**Type** = type;//מה שמתחיל באות גדולה לא מצריך הכרזה כי זה שפת סקריפט

this.**Fuel** = fuel;

this.CalcTankPrice = **calcTankPrice**;//הוספת מתודה

}

function **calcTankPrice**()  {

return this.Full;

}

var car = new Auto('car' , 'benzin');

document.write('My auto is a ' + truck.Type + ".");//גישה לשדה

this.CalcTankPrice();//קריאה למתודה

שיטה נוספת לבניית אובייקטים ע"י יזמים (מתאימה לרוב אם רוצים ליצור עד שלושה):

truck = {type: 'truck' , fuel:'soler'}//בעצם הגדרת תכונות למופע ישר נוצר עם תכונות אלו

document.write('My auto is a ' + truck.type +' and consumes ' + truck.fuel + ".");

הורשה

function ElectrikAuto(fullTank , literPrice, batteryPrice) {  
   **Auto.call(this, fullTank , literPrice)**;  
   this.BatteryPrice = batteryPrice;  
}

Auto.call גורם לקרוא לבנאי האב (ביצוע אתחול הנתונים) כאשר Auto זה האבא, ו-call דומה ברעיון ל-super ב-java. אך יש פרמטר this שהועבר שלא ברור כי במקור האבא מקבל שני פרמטרים, אולי זה כתיב מקובל.

כך אפשר לדרוס פונקציה של האב אצל הבן:

**ElectrikAuto**.**prototype**.calcTankPrice **=** function() {  
   return this.FullTank \* this.LiterPrice \* 0.5;  
}

ElectrikAuto.prototype.calcTankPrice – בעצם אומר לו תציב במתודה שהבן למעשה ירש את המתודה במימוש הזה.

אם יצרנו אובייקטים בשיטה הבאה:

var car = {

model: "BMW",

wheels: 4,

hasSunroof: true

}

for(var prop **in** car){ // כך עוברים על כל האיברים שבאובייקט ולא באמצעות לולאה רגילה שלא ניתן במקרה זה

console.log(prop)//נותן רק את ערך השדה ולא הערך שמקשור לשדה

}

אם רוצים את הערך עצמו, נרשום כך בלולאה:

car["model"] //שולף את הערך שמקושר למשתנה במקרה זה ביאמווי

ואם זה היה משתנה רגיל היינו עושים כך:

var prop = "model"

car[prop]//בלי מרכאות בעצם

**משתנים מקומיים וגלובליים:**

משתנה מקומי מוגדר בתוך פונקציה וגלובלי בכל התכנית (מעל הפונקציה למשל).

אם מעבירים פרמטר לפונקציה או מגדירים בתוכה משתנה בעל שם זהה למשתנה גלובלי, זה שייקבע בתוך הפונקציה זה המקומי (או הפרמטר שהתקבל שגם נחשב מקומי).

דבר חשוב – אם הוגדר משתנה var בבבלוק, עדיין אפשר להשתמש בו מחוץ לבלוק (למשל הוגדר בתוך בלוק If ורוצים להשתמש בו מחוץ לו) אבל חייב שזה יהיה בתוך אותו הסקופ (למשל הסקופ הוא פונקציה מסויימת והוא הוגדר בבלוק if).

אפשר גם להגדיר משתנה ללא var אבל זה הופך לגלובלי (אפילו אם הוגדר בפונקציה, ודורס את הערך של הגלובלי שמחוץ לפונקציה, אבל עדיף שלא לעשות כן).

הוספה/הסרה של אלמנטים ב-JS:

<body>

<**ul** id="mainList">

**<li>Inside the main list.</li>**

</ul>

<ul><li>Control outside the main list</li></ul>

<script>

**var** listItem = document.createElement("li");//יוצרים אלמנט חדש למסמך

**var** textNode = document.**createTextNode**("This is a new list item."); //זה הטקסט שיקושר לאותו אלמנט

**listItem**.appendChild(textNode);//מוסיפים לרשימה הספציפית את התיבה טקסט

**var** mainList = document.getElementById("mainList");

mainList.appendChild(listItem);

</script>

עידכון אלמנטים קיימים (למשל שם):

**<ul id="mainList">**

**<li>No chicken</li>**

**<li>1 chicken</li>**

**<li>2 chicken</li>**

**<li>3 chicken</li>**

**</ul>**

<ul><li>Control outside the main list<li></ul>

<script>

**var** mainList = document.**getElementById**("mainList");

**var** numberOfItems = mainList.**getElementsByTagName**("li").length;//יש כמה מסוג זה

for(**var** i=0; i < **numberOfItems**; i++){

**var** listItem = mainList.getElementsByTagName("li")[i];//מקבלים את האלמנט האיי מסוג זה

// Get the item's text

**var** text = listItem.textContent;

// Replace the word chicken with rabbit

**var** newText = text.replace("chicken", "rabbit");

// Update the list item

listItem.textContent = newText;

}

</script>

הסרה של אלמנט מסויים:

**var** mainList = document.getElementById("mainList");

**var** listItem = mainList.**getElementsByTagName**("li")[0];//האלמנט הראשון מסוג זה

if(listItem) {

mainList.**removeChild**(listItem);

}

**טפסים**

JavaScript מחזיק מערך של כל האלמנטים בדף בתוך אלמנט שנקרא document, ובמערך זה נמצאים כל הטפסים לפי הסדר. הטפסים נקראים forms.

כך ניראה החלק העיקרי ב-HTML של הטופס:

<form name="theForm">

  <input type="text" name="theText">

  <input type="button" value="האם מספיק ארוך?" id="theButton">



לשים לב תגית אחרי תגית ב-HTML מתיישרת באותה שורה, ואם רוצים שורה חדשה עושים br.

וזו הדרך לגישה לשדות (הדרך הראשונה פחות מומלצת כי אם יהיה שינוי מיקום יהיה שינוי אינדקסים):

var x = document.forms[0];

הטופס השני נמצא באינדקס 1 וכך הלאה.

וזה הגישה לשדה בתוך הטופס:

var firstElementInForm = document.forms[0].elements[0];

השיטה הבאה מומלצת והיא לפי שם:

var firstElementName = document.theForm.theText;

theForm – זה שם הטופס בדוגמא שלעיל. theText – זה שם האלמנט הטקסט בטופס.

וזו דוגמא שמתמשת בטופס הזה שנבנה לעיל:

function **theValidator**(){

  // How long is the value of the text field?

  if(**document.theForm.theText**.value.length > 5){

    alert("מספיק ארוך");

  } else {

    alert("קצר מדי");

  }

}

// 1. Listen to the button

// 2. Call the function on click

document.**getElementById**("theButton").**addEventListener**("click", **theValidator**);

הוספנו מאזין ללחיצה על הכפתור והפונקציה נקראת ובודקת את הערך שמופיע בשדה טקסט ובודקת מה אורכו.

**jQuery**

jQuery זו ספרייה של JavaScript שיכולה לייצר הרבה פונקציונליות במעט קוד. לדוגמה, ניתן לייצר אנימציות ואפקטים, AJAX שמאפשר להציג אלמנטים חדשים בדף בלי צורך לרענן את הדפדפן וולידציה של טפסים. רואים את jQuery בכל מקום באינטרנט, מתפריטים שנפתחים כשהעכבר עובר עליהם, תצוגת שקופיות, והשלמה אוטומטית של הטקסט בתיבת טקסט כמו שקורה כשאתם מקלידים משהו לתוך שורת הכתובות בגוגל. לא רק גוגל משתמש ב-jQuery , ולמעשה זו ספרייה פופולרית ביותר בזכות קלות הטעינה, הקלות היחסית של כתיבת הקוד והכמות הגדולה של פלאג-אינים שמוסיפה רבות ליכולות של jQuery.

קודם חשוב לרשום ב-script את השורה הבאה שאומרת להשתמש בספרייה של Jquery

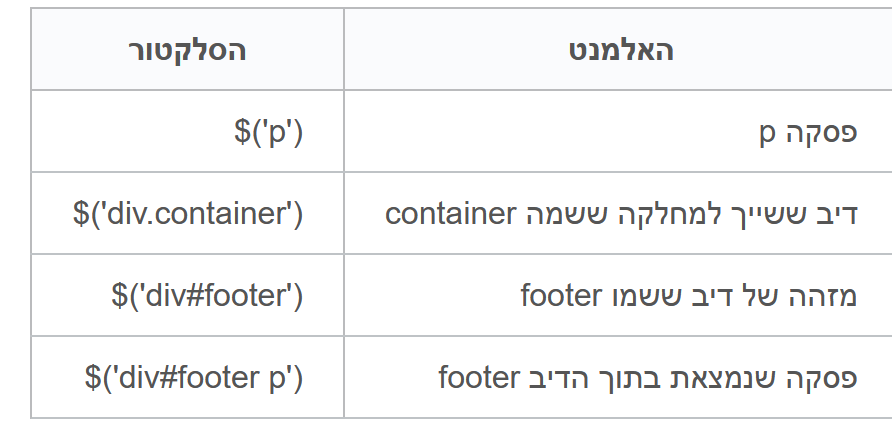
<script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>

**בבסיס תחביר jQuery נמצאים ארבעה מושגי יסוד:**

**סלקטורים**

היתרון הגדול של jQuery שהוא מאפשר לבחור אלמנטים מדף האינטרנט (DOM בשפת ה-JavaScript) באמצעות סלקטורים ששאולים מ-CSS. את הסלקטורים עוטפים בצורה שהופכת אותם למשהו ש-jQuery יכול לעבוד אתו, לאובייקטים של jQyery. לדוגמה, אם רוצים לבחור אלמנט h1 כדי להפעיל עליו את jQuery, צריך לעטוף אותו ב-$(''), והוא הופך לאובייקט ש-jQuery יכול לעבוד אתו. זה נראה כך: $('h1') , ועכשיו h1 הוא אובייקט של jQuery. באופן דומה ניתן לבחור אלמנטים שונים בדף, מחלקות ומזהים.

הטבלה הבאה נותנת מספר דוגמאות:



פונקציה אנונימית, חסרת שם, שמסתירה את הפסקה p, על ידי זה שהיא מפעילה עליה את המתודה hide():

הפונקציה העוטפת שהסימן $ הוא שם תחליפי (alias) ל-jQuery

**(function($){** $('p').hide(); **}) (jQuery) ;**

**אירועים**

קושרים בין הסלקטורים ובין המתודות. דוגמה לאירועים הם: **click**,

 (function($) **{** $('p').on('click', function() { $('h2').hide(); } **}**)(jQuery);

זה פשוט יוצר מאזין (הפונקציה המאזינה רשומה ישר בתוך הכתיב המיוחד), כאשר לוחצים על פסקה אז h2 יוסתר.

ואם נרשום $(this) במקום $('h2') יגרום לאותה יגרום רק לאותה פסקה (שזה ה-this) להיעלם.

רשימת אירועים בולטים:

**()click.**

מתרחש כשהעכבר נמצא על אלמנט, ולאחר שהגולש לחץ על העכבר, ושחרר.

**()dblclick.**

האירוע מתרחש לאחר שהגולש לחץ פעמיים על העכבר.

**()hover.**

כשהעכבר מרחף מעל לאלמנט.

**()mouseenter.**

האירוע מתרחש כשהעכבר נכנס לתוך אלמנט מסוים.

**()mouseleave.**

האירוע מתרחש כשהעכבר יוצא מתוך אלמנט מסוים.

**()keydown.**

כשהמשתמש מתחיל ללחוץ על מקש מסוים.

**()keyup.**

כשהגולש משחרר את המקש המסוים.

**()focusin.**

מתייחס בעיקר לאלמנטים שנמצאים בטפסים ובלינקים. כאשר המשתמש מקליק על האלמנט דוגמת שדה טקסט, או כפתור רדיו.

**()focusout.**

מתייחס בעיקר לאלמנטים שנמצאים בטפסים ובלינקים. כאשר המשתמש יוצא מהאלמנט. לדוגמה, כשהמשתמש יוצא משדה טקסט.

הדרך המומלצת ביותר להאזין לאירועים החל מגרסה 1.7 של **jQuery** היא באמצעות המתודה on, שמחברת לסלקטור (האלמנט הנבחר בדף), את הפונקציה שהוא צריך לבצע בתגובה לאירוע.

$('div#colored-square2').on('click', function () {  
      **$('#circle')**.css('background', 'orange');  
});

מפעילים על האלמנט colored-square את המתודה On (שזה פעולת המאזין למעשה)

מה שיקרה שברגע שנלחץ על colored-square אז הצבע של האלמנט circle ישתנה לכתום.

(אם היינו רושמים this במקום, זה היה משנה את הצבע של colored-square ולא של circle).

**<div>  
   <input type="text" value='הקלד לכאן את הטקסט'/>  
   <input type="submit" value='כפתור להדגמה'/>  
   </div>**

<script>  
(function($){  
   var input = $('input:text'),  
        textInside = input.**val**();//לוקח את הערך שיש באינפוט  
   
   input.on('focusin', function() {//פונקצית האזנה לפוקוס על שדה אינפוט  
       var inputValue = $(this);//  
   
      if (inputValue.val() === textInside)//משווה את הערך שהיה לפני הפוקוס ואחר הפוקוס ובודק אם הטקסט השתנה שם, אם אכן הוא שווה אז  
      {  
      inputValue.attr('value', ' ');//מציב מחרוזות ריקה בווליו  
      }  
   });  
   
   input.on**(**'focusout', function() {//כאשר זה בודק שהפוקוס יצא החוצה

בודק אם המחרוזת ריקה בתיבה אז הוא מחזיר את המחרוזת ברירת המחדל להיכתב שם בתיבה  
      var inputValue = $(this);  
   
      if (inputValue.val() === '')  
      {  
      inputValue.attr('value',textInside);  
      }  
   });  
})(jQuery);

**כיצד לקבל ולשנות תכונות באמצעות jQuery?**

כיצד לקבל ולקבוע טקסט של אלמנט באמצעות ()text?

$('element').text();//קבלת הטקסט הקיים ששייך לאלמנט

$('element').text('set text inside element');//קביעת טקסט חדש עבור האלמנט

הערה- זה לוקח רק את הטקסט ומעתיק ולא עיצוב בולד, צבע וכו'.

**כיצד לקבל ולשנות תכונות של ה-DOM באמצעות ()attr ו-()val?**

כיצד משתמשים ב-()attr. כדי לקבל את ערכו של attribute?

<a href=''>לינק</a>

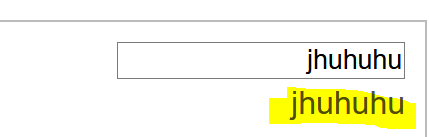
$("שם האלמנט").attr("הערך שנציב לתוכו" , "שם האטריביוט");

$("a").attr("href" , "//www.reshetech/node/120 ");

l

Val יודע לקחת ערכים משדות בטפסים, ולהשתמש בהם או לשנות אותם. בדוגמה הבאה, כל דבר שתקלידו בתיבת הטקסט יופיע מיד מתחת.

(function(){  
כשמקלידים לתוך שדה הטקסט//  
$('input[type="text"]').on('keyup', function() {  
  משתנה text מאחסן את הערך של מה שהוקלד//  
var text = $(this)**.val()**;//שומרים את הטקסט  
המשתנה מוצג לצופה כטקסט //  
$('p#feedback').text(text);//שמים את זה בפסקה של פידבק למטה  
});  
})(jQuery);

סימון צ'קבוקס באופן אוטומטי:

$('input[type="checkbox"]').val(['Checkbox 2']);

כמה דברים חשובים שיש לדעת:

$(this).hide() - hides the current element.

$("p").hide() - hides all <p> elements.

$(".test").hide() - hides all elements with class="test".

$("#test").hide() - hides the element with id="test".

שיטת $ (document) .ready () מאפשרת לנו לבצע פונקציה כאשר המסמך נטען במלואו. אירוע זה כבר מוסבר בפרק התחביר של jQuery.

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Selects** |
| [\*](https://www.w3schools.com/jquery/sel_all.asp) | $("\*") | All elements |
| [#*id*](https://www.w3schools.com/jquery/sel_id.asp) | $("#lastname") | The element with id="lastname" |
| [.*class*](https://www.w3schools.com/jquery/sel_class.asp) | $(".intro") | All elements with class="intro" |
| [.*class,*.*class*](https://www.w3schools.com/jquery/sel_multiple_classes.asp) | $(".intro,.demo") | All elements with the class "intro" or "demo" |
| [*element*](https://www.w3schools.com/jquery/sel_element.asp) | $("p") | All <p> elements |
| [*el1*,*el2*,*el3*](https://www.w3schools.com/jquery/sel_multiple_sel.asp) | $("h1,div,p") | All <h1>, <div> and <p> elements |
|  |  |  |
| [:first](https://www.w3schools.com/jquery/sel_first.asp) | $("p:first") | The first <p> element |
| [:last](https://www.w3schools.com/jquery/sel_last.asp) | $("p:last") | The last <p> element |
| [:even](https://www.w3schools.com/jquery/sel_even.asp) | $("tr:even") | All even <tr> elements |
| [:odd](https://www.w3schools.com/jquery/sel_odd.asp) | $("tr:odd") | All odd <tr> elements |
|  |  |  |
| [:first-child](https://www.w3schools.com/jquery/sel_firstchild.asp) | $("p:first-child") | All <p> elements that are the first child of their parent |
| [:first-of-type](https://www.w3schools.com/jquery/sel_firstoftype.asp) | $("p:first-of-type") | All <p> elements that are the first <p> element of their parent |
| [:last-child](https://www.w3schools.com/jquery/sel_lastchild.asp) | $("p:last-child") | All <p> elements that are the last child of their parent |
| [:last-of-type](https://www.w3schools.com/jquery/sel_lastoftype.asp) | $("p:last-of-type") | All <p> elements that are the last <p> element of their parent |
| [:nth-child(*n*)](https://www.w3schools.com/jquery/sel_nthchild.asp) | $("p:nth-child(2)") | All <p> elements that are the 2nd child of their parent |
| [:nth-last-child(*n*)](https://www.w3schools.com/jquery/sel_nthlastchild.asp) | $("p:nth-last-child(2)") | All <p> elements that are the 2nd child of their parent, counting from the last child |
| [:nth-of-type(*n*)](https://www.w3schools.com/jquery/sel_nthoftype.asp) | $("p:nth-of-type(2)") | All <p> elements that are the 2nd <p> element of their parent |
| [:nth-last-of-type(*n*)](https://www.w3schools.com/jquery/sel_nthlastoftype.asp) | $("p:nth-last-of-type(2)") | All <p> elements that are the 2nd <p> element of their parent, counting from the last child |
| [:only-child](https://www.w3schools.com/jquery/sel_onlychild.asp) | $("p:only-child") | All <p> elements that are the only child of their parent |
| [:only-of-type](https://www.w3schools.com/jquery/sel_onlyoftype.asp) | $("p:only-of-type") | All <p> elements that are the only child, of its type, of their parent |
|  |  |  |
| [parent > child](https://www.w3schools.com/jquery/sel_parent_child.asp) | $("div > p") | All <p> elements that are a direct child of a <div> element |
| [parent descendant](https://www.w3schools.com/jquery/sel_parent_descendant.asp) | $("div p") | All <p> elements that are descendants of a <div> element |
| [element + next](https://www.w3schools.com/jquery/sel_previous_next.asp) | $("div + p") | The <p> element that are next to each <div> elements |
| [element ~ siblings](https://www.w3schools.com/jquery/sel_previous_siblings.asp) | $("div ~ p") | All <p> elements that appear after the <div> element |
|  |  |  |
| [:eq(*index*)](https://www.w3schools.com/jquery/sel_eq.asp) | $("ul li:eq(3)") | The fourth element in a list (index starts at 0) |
| [:gt(*no*)](https://www.w3schools.com/jquery/sel_gt.asp) | $("ul li:gt(3)") | List elements with an index greater than 3 |
| [:lt(*no*)](https://www.w3schools.com/jquery/sel_lt.asp) | $("ul li:lt(3)") | List elements with an index less than 3 |
| [:not(*selector*)](https://www.w3schools.com/jquery/sel_not.asp) | $("input:not(:empty)") | All input elements that are not empty |
|  |  |  |
| [:header](https://www.w3schools.com/jquery/sel_header.asp) | $(":header") | All header elements <h1>, <h2> ... |
| [:animated](https://www.w3schools.com/jquery/sel_animated.asp) | $(":animated") | All animated elements |
| [:focus](https://www.w3schools.com/jquery/sel_focus.asp) | $(":focus") | The element that currently has focus |
| [:contains(*text*)](https://www.w3schools.com/jquery/sel_contains.asp) | $(":contains('Hello')") | All elements which contains the text "Hello" |
| [:has(*selector*)](https://www.w3schools.com/jquery/sel_has.asp) | $("div:has(p)") | All <div> elements that have a <p> element |
| [:empty](https://www.w3schools.com/jquery/sel_empty.asp) | $(":empty") | All elements that are empty |
| [:parent](https://www.w3schools.com/jquery/sel_parent.asp) | $(":parent") | All elements that are a parent of another element |
| [:hidden](https://www.w3schools.com/jquery/sel_hidden.asp) | $("p:hidden") | All hidden <p> elements |
| [:visible](https://www.w3schools.com/jquery/sel_visible.asp) | $("table:visible") | All visible tables |
| [:root](https://www.w3schools.com/jquery/sel_root.asp) | $(":root") | The document's root element |
| [:lang(*language*)](https://www.w3schools.com/jquery/sel_lang.asp) | $("p:lang(de)") | All <p> elements with a lang attribute value starting with "de" |
|  |  |  |
| [[*attribute*]](https://www.w3schools.com/jquery/sel_attribute.asp) | $("[href]") | All elements with a href attribute |
| [[*attribute*=*value*]](https://www.w3schools.com/jquery/sel_attribute_equal_value.asp) | $("[href='default.htm']") | All elements with a href attribute value equal to "default.htm" |
| [[*attribute*!=*value*]](https://www.w3schools.com/jquery/sel_attribute_notequal_value.asp) | $("[href!='default.htm']") | All elements with a href attribute value not equal to "default.htm" |
| [[*attribute*$=*value*]](https://www.w3schools.com/jquery/sel_attribute_end_value.asp) | $("[href$='.jpg']") | All elements with a href attribute value ending with ".jpg" |
| [[*attribute*|=*value*]](https://www.w3schools.com/jquery/sel_attribute_prefix_value.asp) | $("[title|='Tomorrow']") | All elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen |
| [[*attribute*^=*value*]](https://www.w3schools.com/jquery/sel_attribute_beginning_value.asp) | $("[title^='Tom']") | All elements with a title attribute value starting with "Tom" |
| [[*attribute*~=*value*]](https://www.w3schools.com/jquery/sel_attribute_contains_value.asp) | $("[title~='hello']") | All elements with a title attribute value containing the specific word "hello" |
| [[*attribute\**=*value*]](https://www.w3schools.com/jquery/sel_attribute_contains_string_value.asp) | $("[title\*='hello']") | All elements with a title attribute value containing the word "hello" |
|  |  |  |
| [:input](https://www.w3schools.com/jquery/sel_input.asp) | $(":input") | All input elements |
| [:text](https://www.w3schools.com/jquery/sel_input_text.asp) | $(":text") | All input elements with type="text" |
| [:password](https://www.w3schools.com/jquery/sel_input_password.asp) | $(":password") | All input elements with type="password" |
| [:radio](https://www.w3schools.com/jquery/sel_input_radio.asp) | $(":radio") | All input elements with type="radio" |
| [:checkbox](https://www.w3schools.com/jquery/sel_input_checkbox.asp) | $(":checkbox") | All input elements with type="checkbox" |
| [:submit](https://www.w3schools.com/jquery/sel_input_submit.asp) | $(":submit") | All input elements with type="submit" |
| [:reset](https://www.w3schools.com/jquery/sel_input_reset.asp) | $(":reset") | All input elements with type="reset" |
| [:button](https://www.w3schools.com/jquery/sel_input_button.asp) | $(":button") | All input elements with type="button" |
| [:image](https://www.w3schools.com/jquery/sel_input_image.asp) | $(":image") | All input elements with type="image" |
| [:file](https://www.w3schools.com/jquery/sel_input_file.asp) | $(":file") | All input elements with type="file" |
| [:enabled](https://www.w3schools.com/jquery/sel_input_enabled.asp) | $(":enabled") | All enabled input elements |
| [:disabled](https://www.w3schools.com/jquery/sel_input_disabled.asp) | $(":disabled") | All disabled input elements |
| [:selected](https://www.w3schools.com/jquery/sel_input_selected.asp) | $(":selected") | All selected input elements |
| [:checked](https://www.w3schools.com/jquery/sel_input_checked.asp) | $(":checked") | All checked input elements |

תוספות ל-JS:

סקריפטים הכתובים בגוף הדף יופעלו מיד עם טעינת הדף ע"י תוכנת הגלישה. לא תמיד זה מה שאנחנו רוצים. לפעמים, אנחנו רוצים שיתבצעו כאשר נקרא להם, או כאשר מתבצע ארוע כמו סמן העכבר מגיע לנקודה מסוימת. בשביל זה צריך לשים את הסקריפטים תחת חלק ה-Head. סקריפטים שמוגדרים בחלק ה-HEAD יבוצעו רק כאשר יקראו להם.

באופן עקרוני זה יראה כך:

<html>

<head>

<script type="text/javascript">

לעיתים, אם קוראים לאותו סקריפט ממספר דפים, ואז רצוי שהסקריפט יהיה מתואר בקובץ חיצוני, עם סיומת .js

זה יראה בדרך כלל כך:

<html>

<head>

<script src="namescript.js">

Str="123"

N=parseInt(Str)

יציב ל-N את הערך המספרי 123.

האופרטור === **בודק גם ערך וגם סוג**.

"5" == 5 הוא true

"5" === 5 הוא false