

## Syntax

### Operators

- `+` — addition with numbers, concatenation with strings.
- `-` — subtraction.
- `*` — multiplication.
- `/` — division.
- `%` — modulus: get the remainder.
- `++` — add 1 onto the number.
- `--` — subtract 1 from the number.
- `+=` — shorthand add
- `-=` — shorthand subtract
- `*=` — shorthand multiply
- `/=` — shorthand divide

```
var x = 4;
var y = 6;

x++ // 5
y % 4 // 2

x += y; // same as: x = x + y
x -= y; // same as: x = x - y
x *= y; // same as: x = x * y
x /= y; // same as: x = x / y
```

### Variables

```
var name = 'Triceratops'; // String
var numberHorns = 3; // Number
var isHerbivore = true; // Boolean

// Arrays
var carnivores = ['Tyrannosaurus', 'Velociraptor'];

// Objects
var spino = {
  name: 'Spinosaurus',
  isHerbivore: false
}

var isSpinoHerbivore = spino.isHerbivore;
var raptor = carnivores[1];

// Functions
var whatsUp = function (name) {
  alert('What's up ' + name);
}
```

### Logic

- `==` — check if values are the same.
- `===` — check if values are the same, strictly—must be same data type.
- `!=` — check if values are different.
- `!==` — check if values are different, strictly—type is also checked.
- `<=` — less than or equal to.
- `>=` — greater than or equal to.
- `>` — greater than.
- `<` — less than.
- `&&` — and.
- `||` — or.

### Math

- `Math.random()` — generate a random number between 0 and 1.
- `Math.round()` — round the number to an integer.
- `Math.ceil()` — round the number up.
- `Math.floor()` — round the number down.
- `document.documentElement.clientWidth` — the width of the viewport.
- `document.documentElement.clientHeight` — the height of the viewport.

```
// Random between two numbers
var rand = Math.floor(Math.random() * (max - min)) + min;
```

### Strings

- `+` — combine (concatenate) strings together.
- `length` — count how many characters are in the string.
- `toUpperCase()` — convert all the letters to upper case.
- `toLowerCase()` — convert all the letters to lower case.
- `slice(from, to)` — get a chunk of text up to a specific point.
- `substr(from, length)` — get a chunk from the text of a specific length.
- `replace('...', '...')` — change one or more instances to something else.
- `split('...')` — create an array by breaking the string on a specific character.
- `trim()` — remove spaces at the start and end.

### Arrays

- `things[1]` — get a specific item from the array.
- `length` — count how many items are in the array.
- `push(...)` — add to the end of the array.
- `pop()` — remove from the end of the array.
- `unshift(...)` — add to the start of the array.
- `shift()` — remove from the start of the array.
- `delete things[1]` — delete an item from the array.

### If-statements

```
if (planet == 'Venus') {
  // True path
} else {
  // False path
}

if (planet == 'Venus' && isSecond) {

}
```

### Loops

```
var planets = ['Mercury', 'Venus', 'Earth', 'Mars'];
var i = 0;
var totalPlanets = planets.length;

for (i = 0; i < totalPlanets; i++) {
  // For-loop
}

planets.forEach(function (item) {
  // For-each loop
});
```

### Switches

```
var planet = 'Venus';

switch (planet) {
  case 'Mercury' :
    break;

  case 'Venus' :
    break;

  case 'Mars' :
    break;
}
```

## jQuery & HTML

### Selections & events

```
var $dinos = $('<div>');
var $trex = $('#trex');
var $action = $('[data-action]');
var $radio = $('[name="diet"]:checked');

// Including <> in the jQuery selector
// will make a brand new element
var $newDiv = $('<div>');

$trex.on('click', function (e) {
  // Click directly on #trex
});

$dinos.on('click', 'li', function (e) {
  // Click on any <li> within .dinos
});

$('form').on('submit', function (e) {
  // Stop the default action
  e.preventDefault();
});
```

**Common events:** click, focus, blur, mousedown, mouseover, mouseout, change, keydown, submit, animationend, transitionend & [many, many more...](#)

### Manipulations

attr(attribute) — get an attribute value.  
attr(attribute, value) — change an attribute value.  
html() — get the entire HTML content of an element.  
html(value) — change the entire HTML content of an element.  
addClass(class) — add a class to an element.  
removeClass(class) — remove a class from an element.  
toggleClass(class) — toggle a class on/off.  
hasClass(class) — check if an element has a specific class.  
val() — get the form input's value.  
val(value) — set the form input's value; val('') would clear the input.  
append(value) — add new HTML inside the element, after what's there.  
prepend(value) — add new HTML inside the element, before what's there.  
before(value) — add new HTML outside the element, before it.  
after(value) — add new HTML outside the element, after it.  
remove() — delete the element from the page.  
show() — set an element's display property to make it visible—use with caution.  
hide() — set display: none—use with caution.

### Traversals

each(function) — loop over all the selected elements.  
filter(selector) — filter the selected elements by selector.  
find(selector) — find a specific element within another element.  
parent() — get the parent element of the current element.  
parents(selector) — find a parent element.  
children(selector) — find specific child elements.  
siblings(selector) — find specific siblings.  
eq(index) — get the element by its index.  
get() — get the raw Javascript element.

```
<ul>
  <li class="pterasaur">Quetzalcoatlus</li>
  <li class="dinosaur">Apatosaurus</li>
  <li class="plesiosaur">Elasmosaurus</li>
</ul>
```

```
$('li').each(function () {
  if ($(this).hasClass('dinosaur')) {
    $(this).show();
  }
});
```

## Sample programs

### Toggle class on list

```
<ul>
  <li>Titanosaurus</li>
  <li>Apatosaurus</li>
```

### Generate a list

```
<ul class="dinos"></ul>
```

### Add class to parent

```
<main>
  <article>
    <button id="click-me">Click me!</button>
```

```
<li>Brontosaurus</li>
<li>Brachiosaurus</li>
</ul>
```

```
.is-clicked {
  background-color: limegreen;
}
```

```
$('#ul').on('click', 'li', function (e) {
  $(this).toggleClass('is-clicked');
});
```

```
var dinos = ['Stegosaurus', 'Ankylosaurus', 'Triceratops']

dinos.forEach(function (dino) {
  var $li = $('<li>');
  $li.html(dino);
  $('body').append($li);
});
```

```
</article>
</main>
```

```
$('#click-me').on('click', function (e) {
  $(this).parents('main').addClass('is-clicked');
});
```

### Keyboard events

```
$('#html').on('keydown', function (e) {
  var $theKey = $('strong');
  $theKey.html(e.keyCode);
  $('body').append($theKey);
});
```

### Random width & height

```
<div class="rect"></div>
```

```
var randWidth = Math.random() * 256;
var randHeight = Math.random() * 256;

$('.rect').css('width', randWidth);
$('.rect').css('height', randHeight);
```

### Trigger animation

```
<div class="ball"></div>
```

```
.ball {
  background-color: red;
  animation-duration: 1s;
  animation-timing-function: ease-in-out;
}

@keyframes change-colour {

  0% {
    background-color: red;
  }

  100% {
    background-color: green;
  }

}
```

```
$('.ball').css('animation-name', 'change-colour');
```