Bootstrap

Quick start

1. <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">

1. <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
2. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>
3. <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>

Containers

- Containers are the most basic layout element in Bootstrap and are required when using our default grid system. Choose from a responsive, fixed-width container (meaning its max-width changes at each breakpoint) or fluid-width (meaning it’s 100% wide all the time).

- Variants:
  - Container
  - Container-fluid

Responsive breakpoints

- // Extra small devices (portrait phones, less than 576px)
// No media query for `xs` since this is the default in Bootstrap

- // Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

- // Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

- // Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

- // Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }

Responsive breakpoints

- @include media-breakpoint-only(xs) { ... }
- @include media-breakpoint-only(sm) { ... }
- @include media-breakpoint-only(md) { ... }
- @include media-breakpoint-only(lg) { ... }
- @include media-breakpoint-only(xl) { ... }

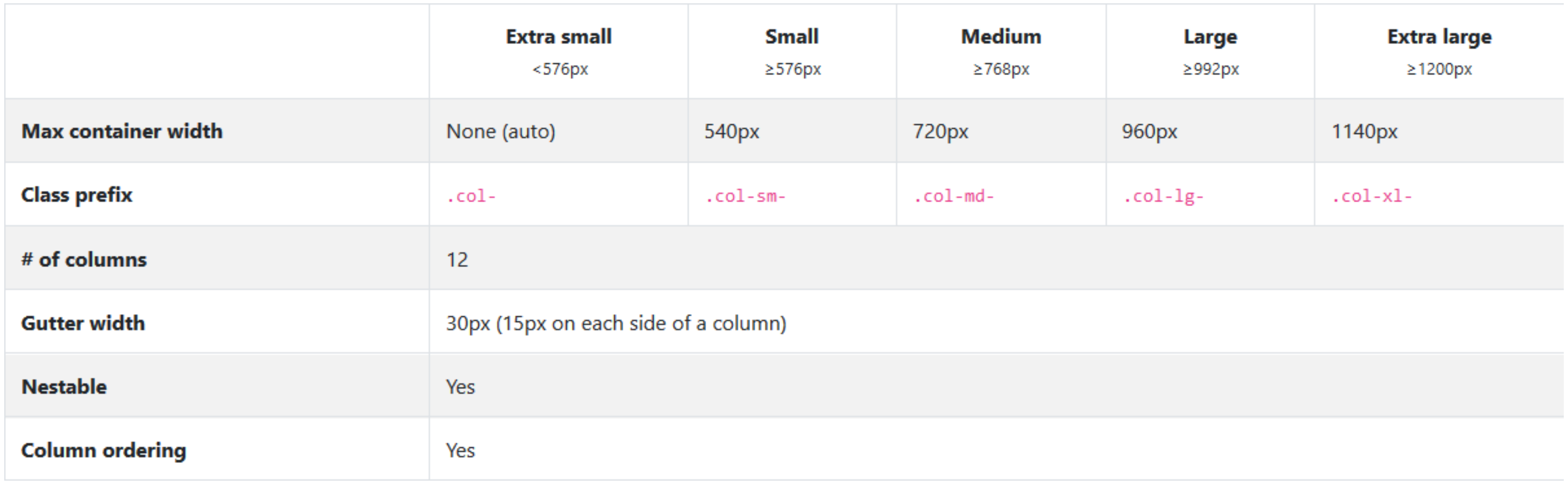
Grid system

```html
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

Grid system

- Containers provide a means to center and horizontally pad your site's contents. Use .container for a responsive pixel width or .container-fluid for width: 100% across all viewport and device sizes.
- Rows are wrappers for columns. Each column has horizontal padding (called a gutter) for controlling the space between them. This padding is then counteracted on the rows with negative margins. This way, all the content in your columns is visually aligned down the left side.
- In a grid layout, content must be placed within columns and only columns may be immediate children of rows.
- Thanks to flexbox, grid columns without a specified width will automatically layout as equal width columns. For example, four instances of .col-sm will each automatically be 25% wide from the small breakpoint and up. See the auto-layout columns section for more examples.
- Column classes indicate the number of columns you'd like to use out of the possible 12 per row. So, if you want three equal-width columns across, you can use .col-4.
- Column widths are set in percentages, so they're always fluid and sized relative to their parent element.
- Columns have horizontal padding to create the gutters between individual columns, however, you can remove the margin from rows and padding from columns with .no-gutters on the .row.
- To make the grid responsive, there are five grid breakpoints, one for each responsive breakpoint: all breakpoints (extra small), small, medium, large, and extra large.
- Grid breakpoints are based on minimum width media queries, meaning they apply to that one breakpoint and all those above it (e.g., .col-sm-4 applies to small, medium, large, and extra large devices, but not the first xs breakpoint).
- You can use predefined grid classes (like .col-4) or Sass mixins for more semantic markup.

Grid options



	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

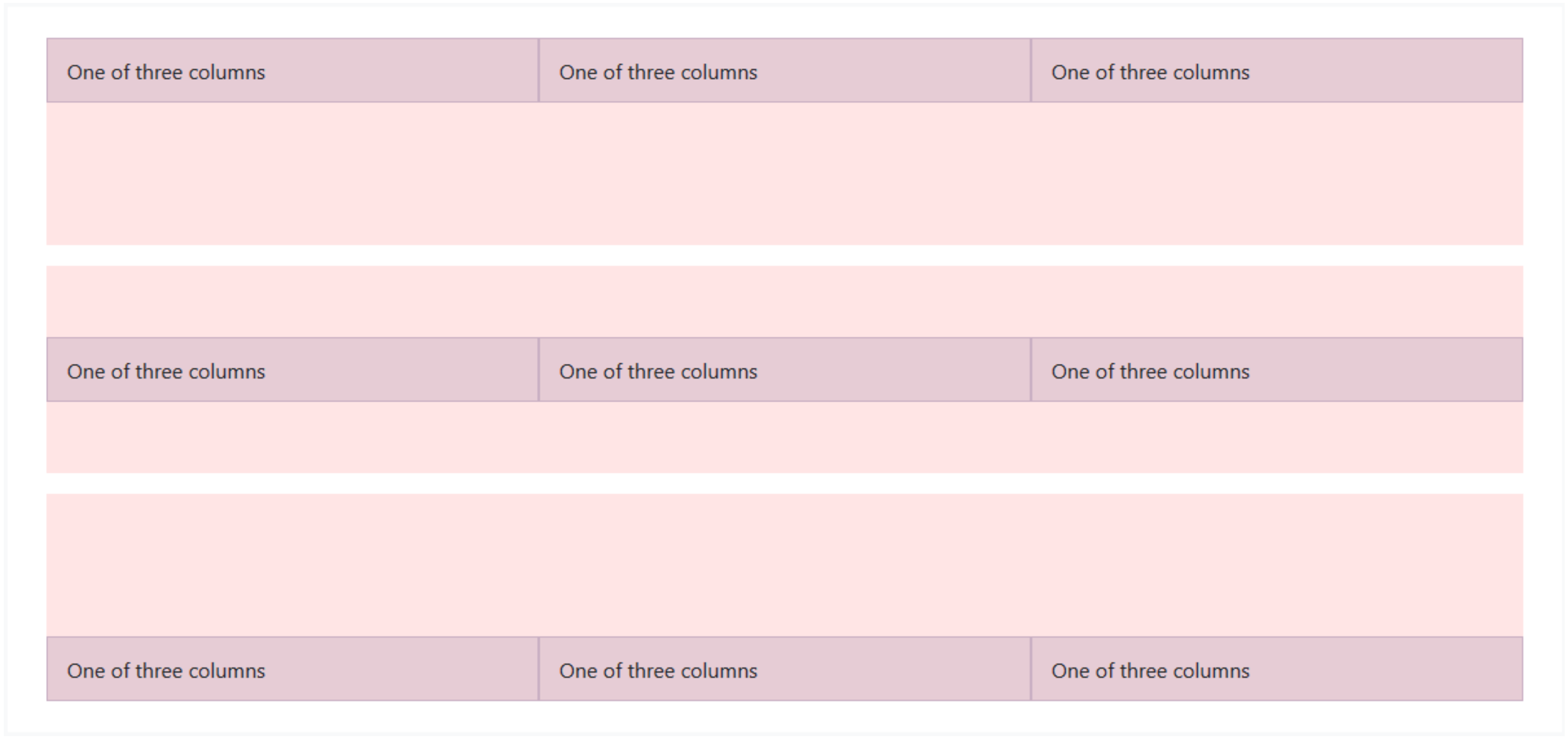
Equal-width multi-row

```
<div class="container">
  <div class="row">
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="w-100"></div>
    <div class="col">col</div>
    <div class="col">col</div>
  </div>
</div>
```

Vertical alignment

```
<div class="container">
  <div class="row align-items-start">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
  <div class="row align-items-center">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
  <div class="row align-items-end">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
</div>
```

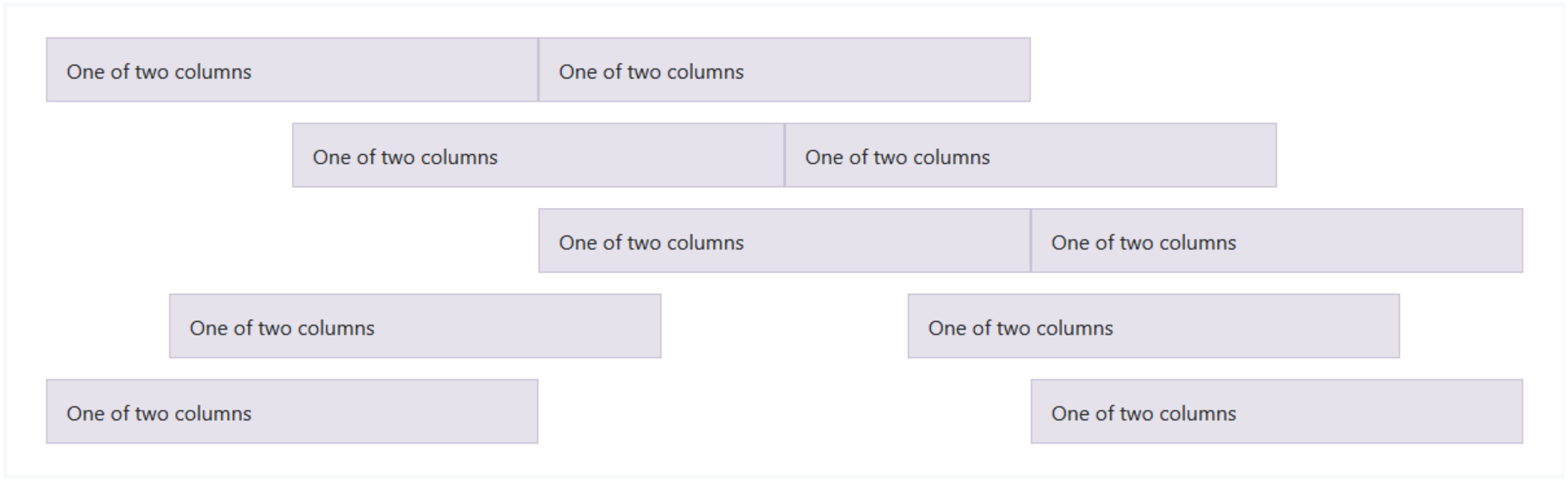
Vertical alignment



Horizontal alignment

```html
<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-end">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-around">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-between">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
</div>
```

Horizontal alignment



Useful links

- https://getbootstrap.com/docs/4.2/components/alerts/
- https://getbootstrap.com/docs/4.2/examples/album/
- https://getbootstrap.com/docs/4.2/examples/checkout/
- https://getbootstrap.com/docs/4.2/examples/pricing/

Connect

- The <script> Tag
  - Internal JavaScript in <head>
  - Internal JavaScript in <body>
  - External JavaScript in <head>
    - External file
    - External References
  - External JavaScript in <body>
    - External file
    - External References

- Internal JavaScript Functions and Events

Output data

- Console
    - Log
    - Info
    - Other commands

- Alert
- Prompt
- innerHTML
- document.write()

Statements

```
var x, y=10, z;    // Statement 1
x = 5;             // Statement 2
y = 6;             // Statement 3
z = x + y;         // Statement 4
```

Literals

Numbers:
- 0.33
- 0.5
- 0.7
- 1

Strings
- ‘Jason’
- “Bourne”

Comments

- // Ordinary comments
- /* multi-
  line
  */ comments

Operators

Simple:

- ‘+’ – Addition
- ‘-’ – Subtraction
- ‘*’ – Multiplication
- ‘/’ – Division
- ‘=’ – Assignment
- ‘++’ – Increment
- ‘--’ – Decrement
- ‘%’ – Remainder

Complex assignment:

- ‘+=’
- ‘-=’

Comparison operators

Severe inequality:

- ==
- !=
- >
- <
- <=
- >=

Strict inequality:

- ===
- !==

Functions

A JavaScript function is a block of code designed to perform a particular task.

```
function Shmunction(paaram1, param2) {
  return param1 * param2;    // The function returns the product of p1 and p2
}
```

Local Variables

```
// code here can NOT use carName
function myFunction() {
    var carName = "Volvo";
    // code here CAN use carName
}
// code here can NOT use carName
```

Object

```
var person = {
  firstName:"John",
  lastName:"Doe",
  age:50,
  eyeColor:"blue"
};
```

Accessing Object Properties

- objectName.propertyName
- objectName["propertyName"]

Example:

```
console.log(person.lastName);

Var a=‘name’

console.log(person.[a]);
```

Methods on object

```
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

this

- In a function definition, this refers to the "owner" of the function.
- In the example above, this is the person object that "owns" the fullName function.
- In other words, this.firstName means the firstName property of this object.

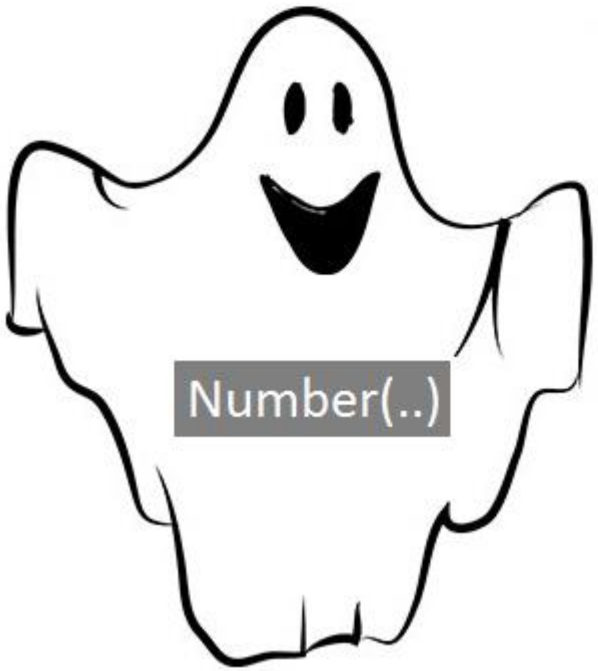
Generate object

```
var x = new String();        // Declares x as a String object

var y = new Number();        // Declares y as a Number object

var z = new Boolean();       // Declares z as a Boolean object
```

Type conversion



Type conversion

JavaScript provides several different options for forcing casts between types. For example:

1. var a = " 42";
2. var b =Number(a);
3. console.log (a); / / " 42"
4. console.log (b);/ / 42
5. console.log (String(b));/ / " 42"

Switch

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

Loops

```
for (var i = 0; i < cars.length; i++) {
 text += cars[i] + "<br>";
}
while(i < cars.length)
 {
 text += cars[i] + "<br>";
 i++;
}
do {
 text += cars[i] + "<br>";
 i++;
}
while(i < cars.length)
```

Date

- new Date()
- new Date(year, month, day, hours, minutes, seconds, milliseconds)
- new Date(milliseconds)
- new Date(date string)
- new Date("October 13, 2014 11:13:00") // Datestring
- new Date(0) // ms by 01.01.1970

Math

- Math.round()
- Math.pow()
- Math.sqrt()
- Math.abs()
- Math.ceil()
- Math.floor()
- Math.sin()
- Math.cos()
- Math.min() and Math.max()
- Math.random()

Math Properties

- Math.E // returns Euler's number
- Math.PI // returns PI
- Math.SQRT2 // returns the square root of 2
- Math.SQRT1-2 // returns the square root of 1/2
- Math.LN2 // returns the natural logarithm of 2
- Math.LN10 // returns the natural logarithm of 10
- Math.LOG2E // returns base 2 logarithm of E
- Math.LOG10E // returns base 10 logarithm of E

Exception handling

```
function myFunction() {
  var message, x;
  message = document.getElementById("p01");
  message.innerHTML = "";
  x = document.getElementById("demo").value;
  try {
    if(x == "") throw "empty";
    if(isNaN(x)) throw "not a number";
    x = Number(x);
    if(x < 5) throw "too low";
    if(x > 10) throw "too high";
  }
  catch(err) {
    message.innerHTML = "Input is " + err;
  }
}
```

Finally

```
function myFunction() {
  var message, x;
  message = document.getElementById("p01");
  message.innerHTML = "";
  x = document.getElementById("demo").value;
  try {
    if(x == "") throw "is empty";
    if(isNaN(x)) throw "is not a number";
    x = Number(x);
    if(x > 10) throw "is too high";
    if(x < 5) throw "is too low";
  }
  catch(err) {
    message.innerHTML = "Error: " + err + ".";
  }
  finally {
    document.getElementById("demo").value = "";
  }
}
```

Error structure

Property	Description
name	Sets or returns an error name
message	Sets or returns an error message (a string)

Error Name	Description
EvalError	An error has occurred in the eval() function
RangeError	A number "out of range" has occurred
ReferenceError	An illegal reference has occurred
SyntaxError	A syntax error has occurred
TypeError	A type error has occurred
URIError	An error in encodeURI() has occurred

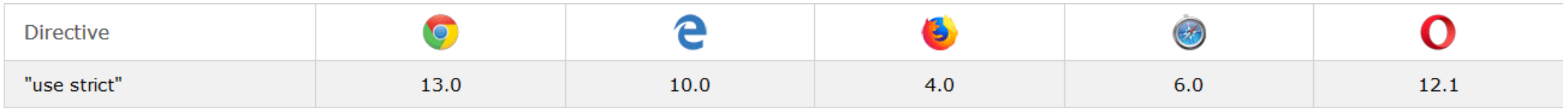
‘Strict’ Mode

The "use strict" directive was new in ECMAScript version 5.

It is not a statement, but a literal expression, ignored by earlier versions of JavaScript.

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".

With strict mode, you can not, for example, use undeclared variables.



Directive					
"use strict"	13.0	10.0	4.0	6.0	12.1

Events

1. Mouse event
2. Window object event
3. Keyboard event
4. Form event
5. Clipboard events
6. Drag & drop event
7. CSS event

Mouse event

click
dblclick
contextmenu
mouseover
mousedown
mouseup
mousemove
onwheel

Window object event

DOMContentLoaded
load
beforeunload/unload
resize
error

Form event

blur
change
focus
invalid
select
submit
input