MP1 Mixed Bag

September 18th, 2017

CSS

Centering

Setup

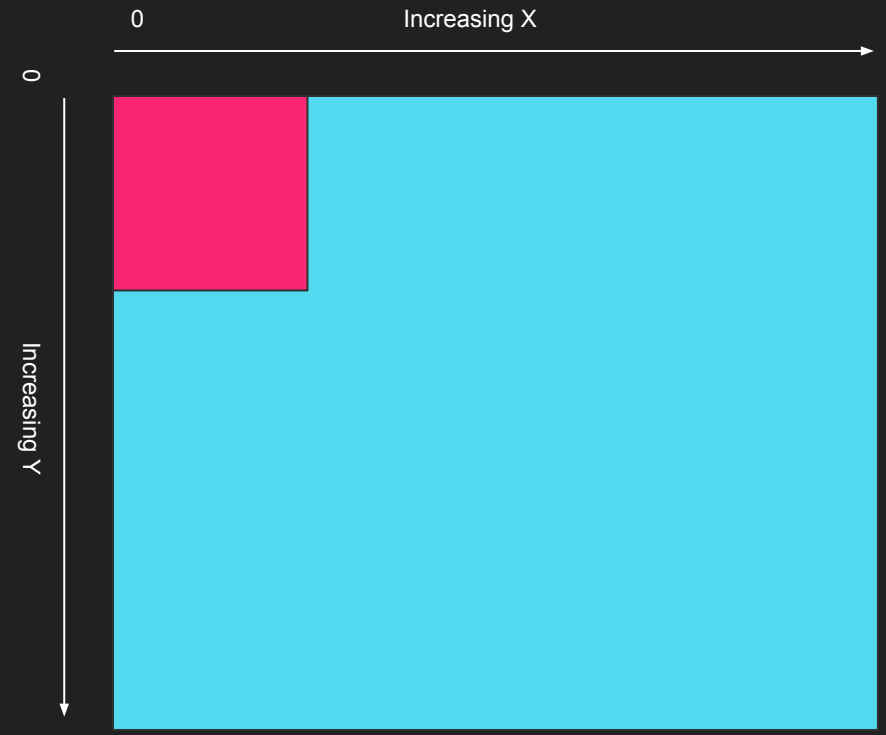
- .outer is the parent element of .inner. We want to center .inner inside of .outer, regardless of the content size of .inner

```
.outer {

}
```

```
.inner {

}
```



position: absolute;

- Remember from lecture that “position: absolute;” means the element becomes positioned *relative to the first non-static parent element.*
- Because .outer is still static, .inner is now absolutely positioned relative to the whole page, instead of .outer

```
.outer {

}
```

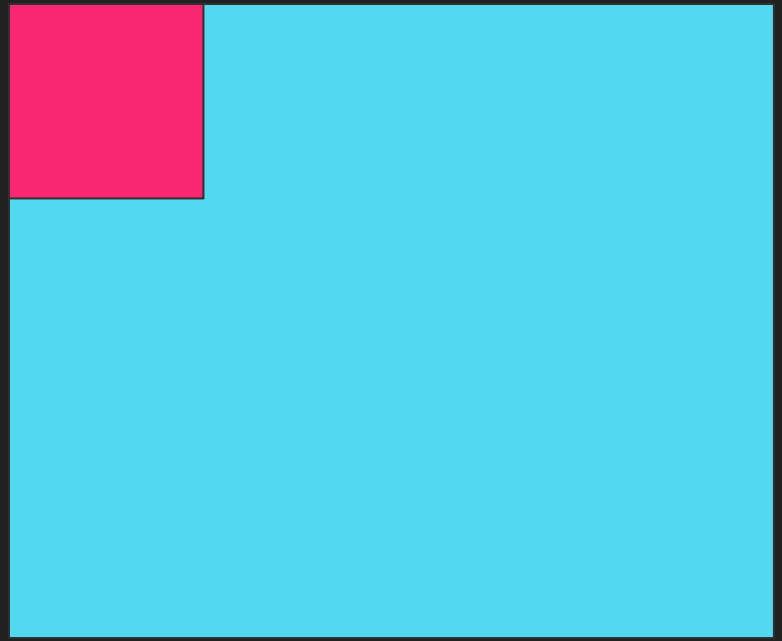
```
.inner {
        position: absolute;
}
```

position: relative;

- By adding “position: relative;” to .outer, we can now position .inner relative to its parent element, .outer

```
.outer {
        position: relative;
}
```

```
.inner {
        position: absolute;
}
```

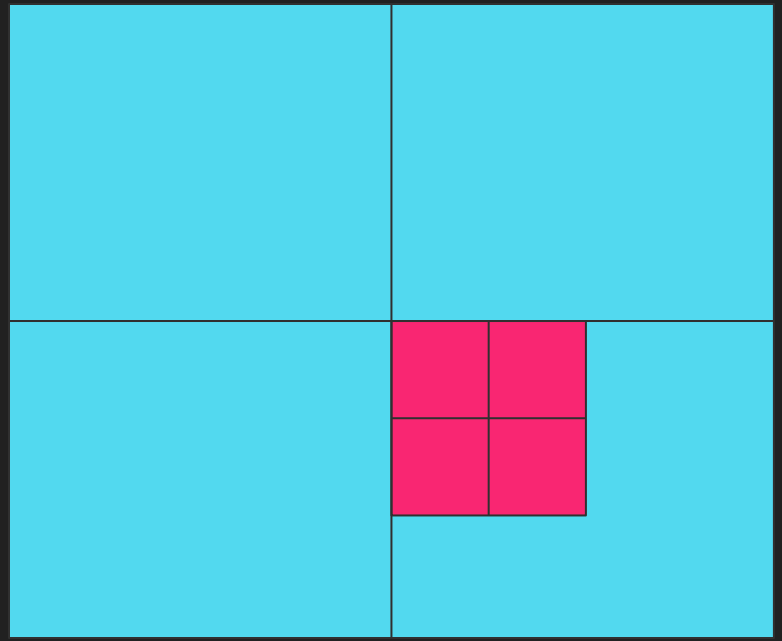


top: 50%; left: 50%;

- We now want to position .inner. The properties top, right, bottom, and left all position an element *based on whatever its first non-static parent container's dimensions are.*
- So, for example, "top: 50%;" moves .inner *down by 50% of .outer height,* and "left: 50%;" moves it over similarly.
- Now, we've successfully positioned the top-left corner of .inner in the middle!! But it's still off-center...

```
.outer {
        position: relative;
}
```

```
.inner {
        position: absolute;
        top: 50%;
        left: 50%;
}
```

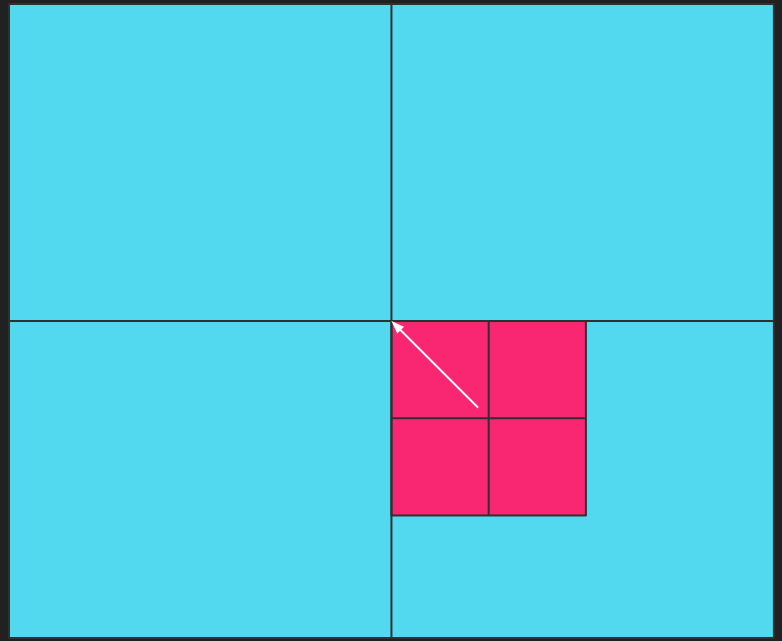


The problem

- To fix .inner being off-center, we need to understand why this happens. Using absolute positioning like this moves the *top-left corner* of the element to the very middle of the page.
- If we want to move the very center of .inner in the very center of .outer, we need to nudge .inner to the left by exactly half of .inner's width and exactly half of .inner's height

```
.outer {
        position: relative;
}
```

```
.inner {
        position: absolute;
        top: 50%;
        left: 50%;
}
```

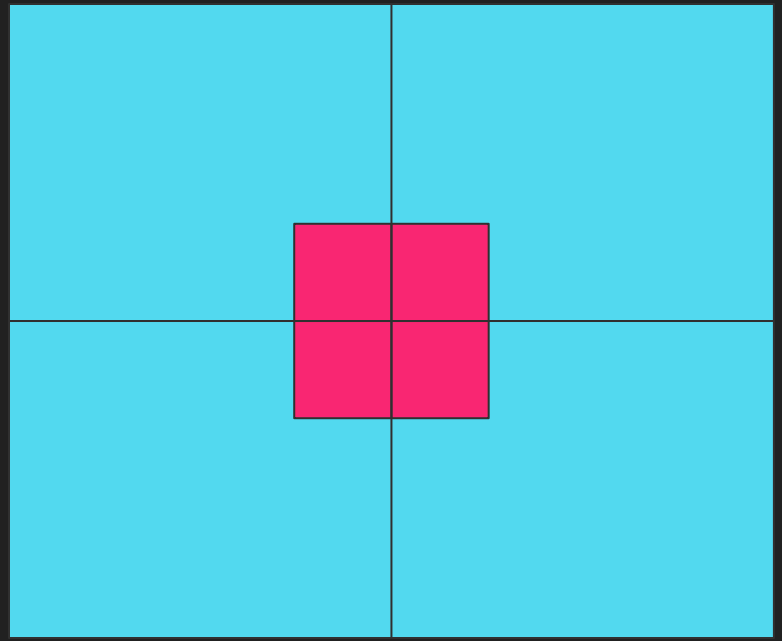


transform: translate(-50%, -50%);

- transform: translate(x, y); moves an element *relative to its OWN height and width. (NOT THE PARENT'S)*
- Because we want to shift .inner up and to the left, we want to move negatively along the x and y axis by 50% of its own width and height.
- Thus, we apply transform; translate(-50%, -50%); to .inner.

```
.outer {
        position: relative;
}
```

```
.inner {
        position: absolute;
        top: 50%;
        left: 50%;
        transform:
translate(-50%, -50%);
}
```



Summary

- Know how and when to use position: absolute; and position: relative;
- Top, right, bottom, left properties shift an element's position *relative to its parent container's dimensions*
- Transform: translate(x, y) shifts an element's position *relative to its own dimensions*

Flexbox

What is Flexbox?

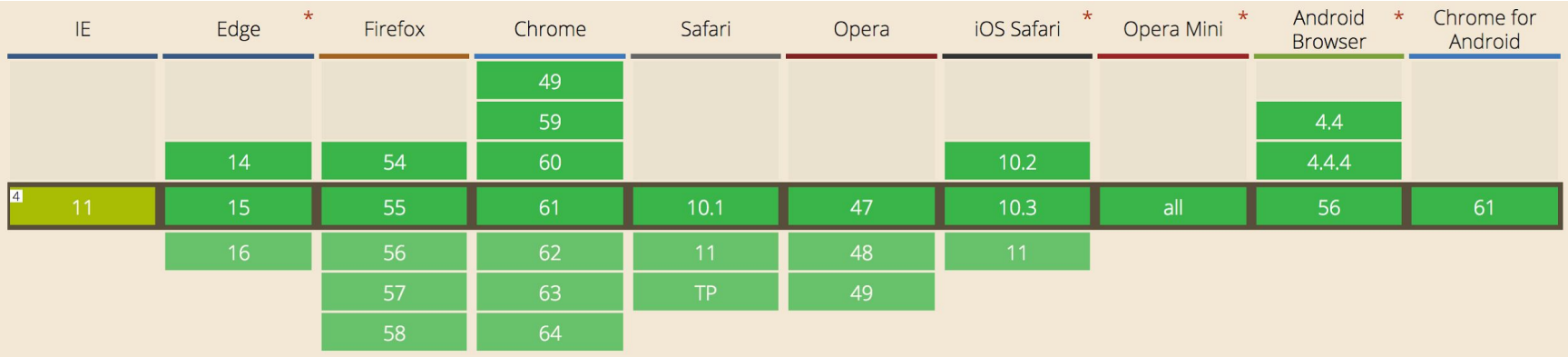
“Flexible Box Layout”

Efficient way to lay out content dynamically

Flex containers expands its items to fit the items in the available space

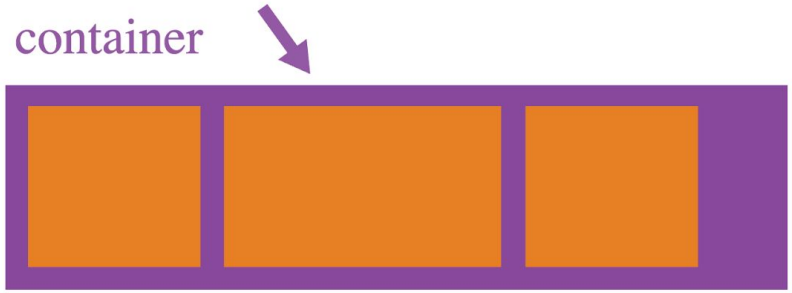
Direction-agnostic

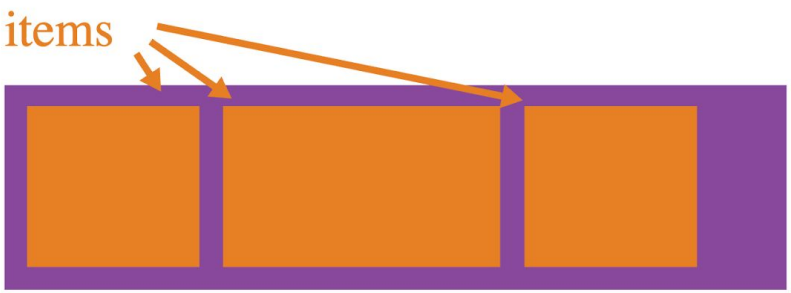
Compatibility



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			59					4.4	
	14	54	60			10.2		4.4.4	
4 11	15	55	61	10.1	47	10.3	all	56	61
	16	56	62	11	48	11			
		57	63	TP	49				
		58	64						

Terminology

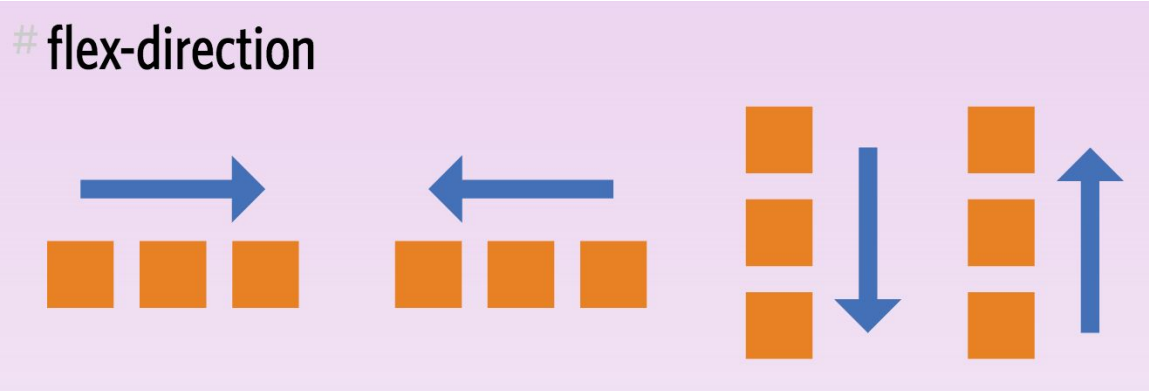





```
.container {
  display: flex; /* or inline-flex */
}
```

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

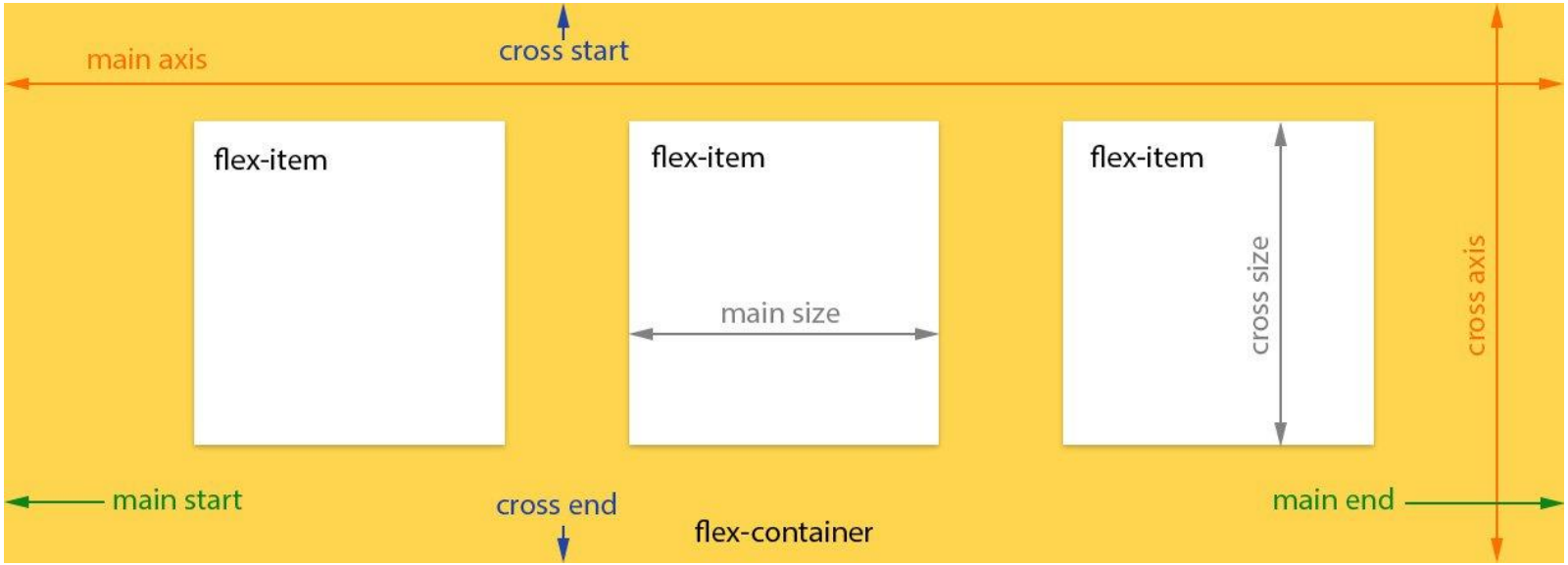
Terminology




```
.container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

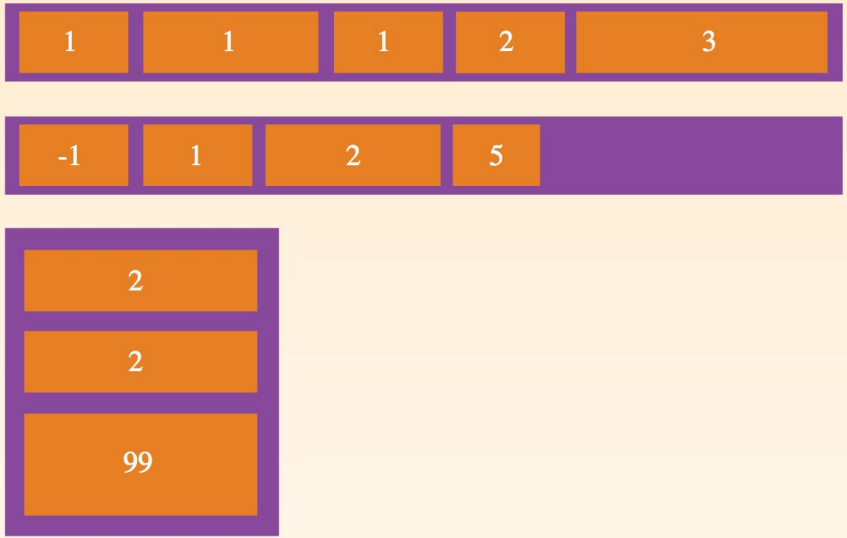
Terminology



https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties

Terminology

order

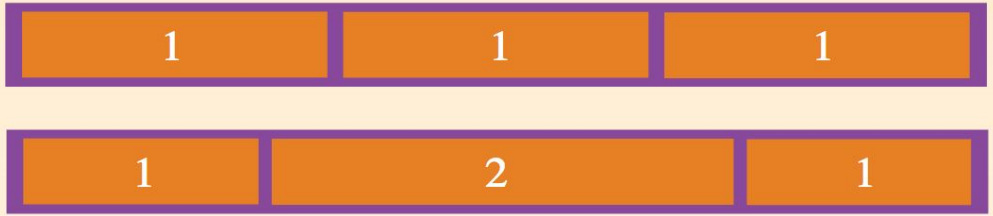



```
.item {
  order: <integer>;
}
```

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Terminology

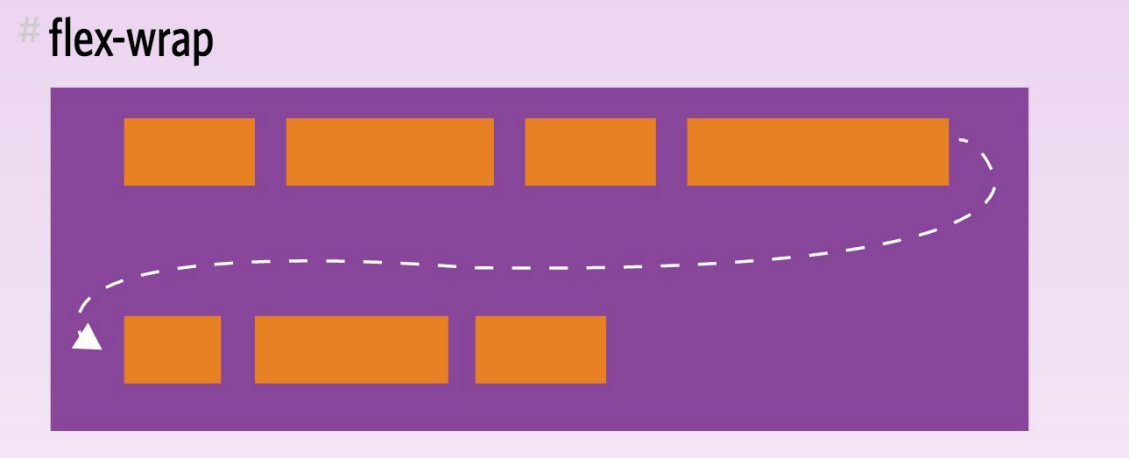
flex-grow




```
.item {
  flex-grow: <number>; /* default 0 */
}
```

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Terminology

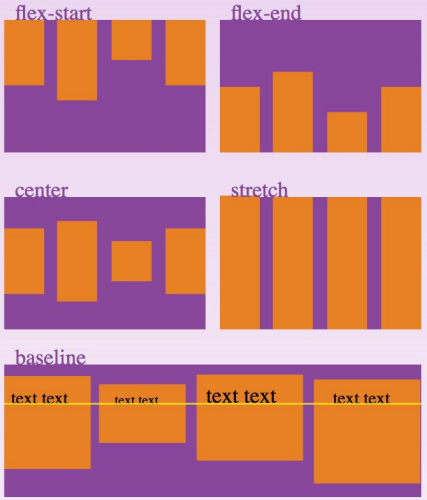



```
.container{
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

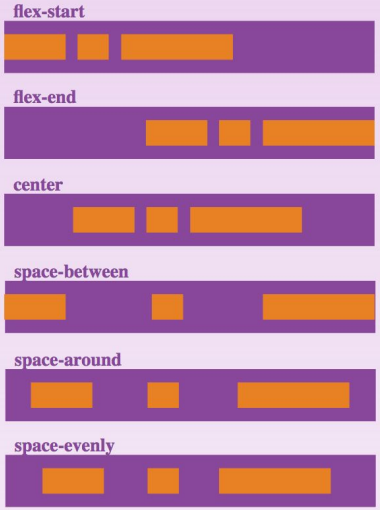
https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Terminology

align-items



justify-content



https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Perfect Centering

```
.parent {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Further Reading

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox#Why_Flexbox

https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties

https://codepen.io/justd/pen/yydezN/

Responsive Breakpoints

Responsivity

Mobile-first world

MP1: 1920x1080, 1366x768, 1280x720, 1024x768

Flexbox

Media Queries

```
@media (query) {
  /* CSS Rules used when query matches */
}
```

https://developers.google.com/web/fundamentals/design-and-ui/responsive/

Responsivity

Parameters
min-width
Rules applied for any browser width greater than the value defined in the query.
max-width
Rules applied for any browser width less than the value defined in the query.
min-height
Rules applied for any browser height greater than the value defined in the query.
max-height
Rules applied for any browser height less than the value defined in the query.
orientation=portrait
Rules applied for any browser where the height is greater than or equal to the width.
orientation=landscape
Rules for any browser where the width is greater than the height.

```
@media (min-width: 360px) {
  body {
    font-size: 1.0em;
  }
}
```

https://developers.google.com/web/fundamentals/design-and-ui/responsive/

Responsivity

Not recommended—fixed width

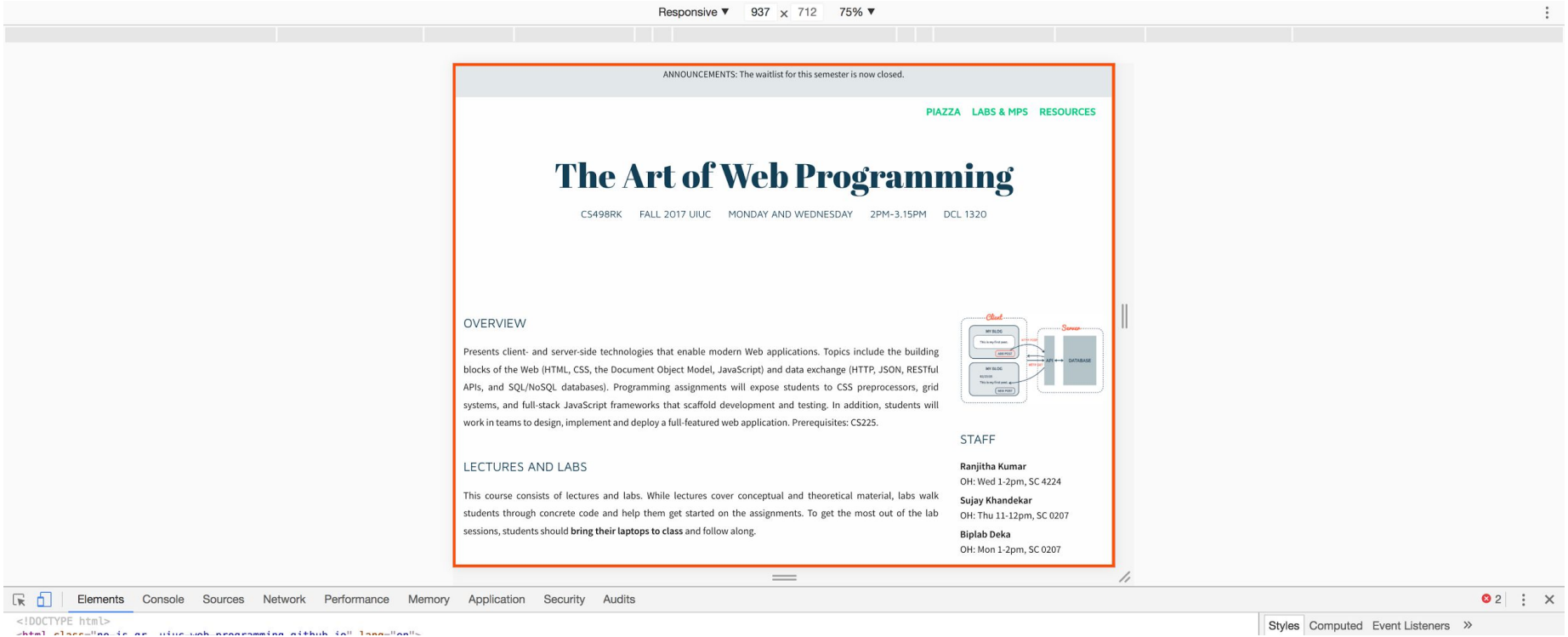
```
div.fullWidth {
  width: 320px;
  margin-left: auto;
  margin-right: auto;
}
```

Recommended—responsive width

```
div.fullWidth {
  width: 100%;
}
```

https://developers.google.com/web/fundamentals/design-and-ui/responsive/

Chrome Inspector



JAVASCRIPT

ES6

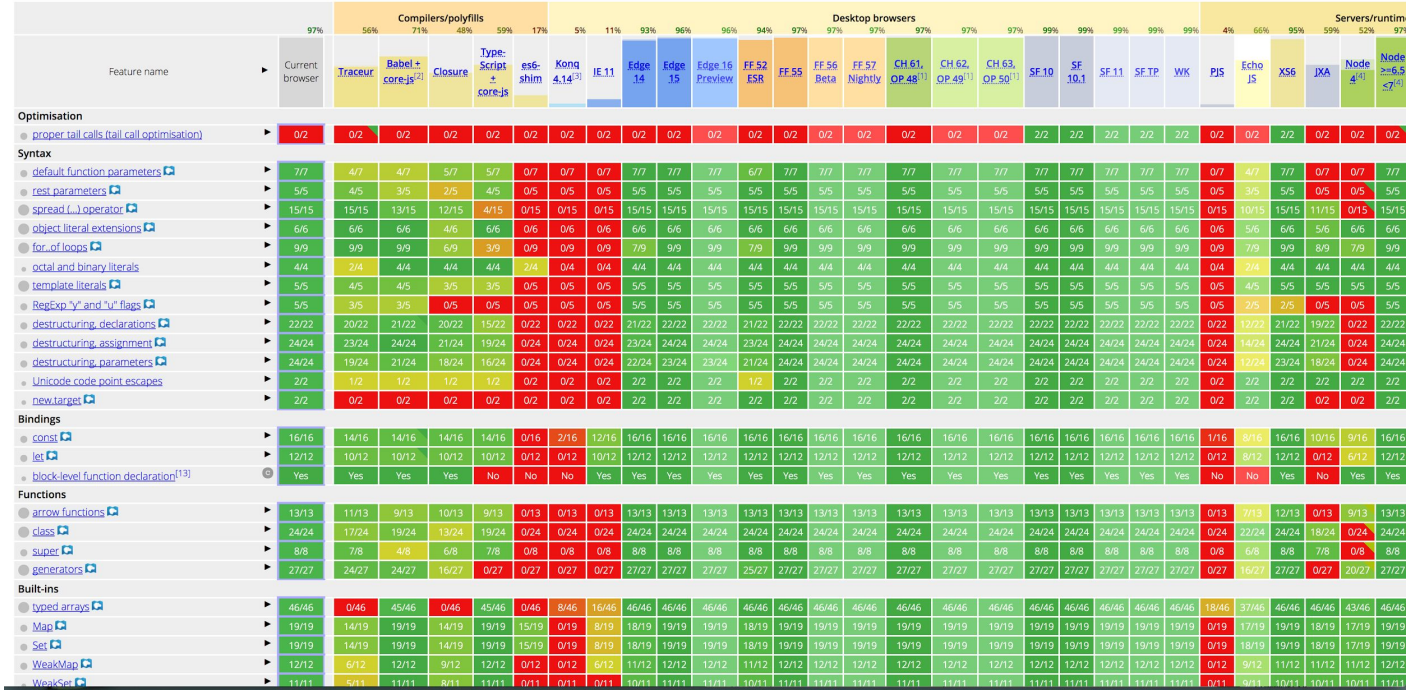
Why ES6?

EcmaScript 2015, or ES6

More Native Functionality & Methods

Less headaches

Compatibility



Feature name	Current browser	Compilers/polyfills: Traceur	Babel + core-js[2]	Closure	TypeScript + core-js	es6-shim	Desktop browsers: Kong 4.14[3]	IE 11	Edge 14	Edge 15	Edge 16 Preview	FF 52 ESR	FF 55	FF 56 Beta	FF 57 Nightly	CH 61, OP 48[1]	CH 62, OP 49[1]	CH 63, OP 50[1]	SF 10	SF 10.1	SF 11	SF TP	WK	Servers/runtim: PJS	Echo JS	XS6	jXA	Node 4[4]	Node >=6.5 <7[4]
	97%	56%	71%	48%	59%	17%	5%	11%	93%	96%	96%	94%	97%	97%	97%	97%	97%	97%	99%	99%	99%	99%	99%	4%	66%	95%	59%	52%	97%
Optimisation																													
proper tail calls (tail call optimisation)	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	0/2	0/2	2/2	0/2	0/2	0/2
Syntax																													
default function parameters	7/7	4/7	4/7	5/7	5/7	0/7	0/7	0/7	7/7	7/7	7/7	6/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	0/7	4/7	7/7	0/7	0/7	7/7
rest parameters	5/5	4/5	3/5	2/5	4/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	3/5	5/5	0/5	0/5	5/5
spread (...) operator	15/15	15/15	13/15	12/15	4/15	0/15	0/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	0/15	10/15	15/15	11/15	0/15	15/15
object literal extensions	6/6	6/6	6/6	4/6	6/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	0/6	5/6	6/6	5/6	6/6	6/6
for..of loops	9/9	9/9	9/9	6/9	3/9	0/9	0/9	0/9	7/9	9/9	9/9	7/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	0/9	7/9	9/9	8/9	7/9	9/9
octal and binary literals	4/4	2/4	4/4	4/4	4/4	2/4	0/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	0/4	2/4	4/4	4/4	4/4	4/4
template literals	5/5	4/5	4/5	3/5	3/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	4/5	5/5	5/5	5/5	5/5
RegExp "y" and "u" flags	5/5	3/5	3/5	0/5	0/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	2/5	2/5	0/5	0/5	5/5
destructuring, declarations	22/22	20/22	21/22	20/22	15/22	0/22	0/22	0/22	21/22	22/22	22/22	21/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	0/22	12/22	21/22	19/22	0/22	22/22
destructuring, assignment	24/24	23/24	24/24	21/24	19/24	0/24	0/24	0/24	23/24	24/24	24/24	23/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	0/24	14/24	24/24	21/24	0/24	24/24
destructuring, parameters	24/24	19/24	21/24	18/24	16/24	0/24	0/24	0/24	22/24	23/24	23/24	21/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	0/24	12/24	23/24	18/24	0/24	24/24
Unicode code point escapes	2/2	1/2	1/2	1/2	1/2	0/2	0/2	0/2	2/2	2/2	2/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	2/2	2/2	2/2	2/2
new.target	2/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	2/2	0/2	0/2	2/2
Bindings																													
const	16/16	14/16	14/16	14/16	14/16	0/16	2/16	12/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	1/16	8/16	16/16	10/16	9/16	16/16
let	12/12	10/12	10/12	10/12	10/12	0/12	0/12	10/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	0/12	8/12	12/12	0/12	6/12	12/12
block-level function declaration[13]	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes
Functions																													
arrow functions	13/13	11/13	9/13	10/13	9/13	0/13	0/13	0/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	0/13	7/13	12/13	0/13	9/13	13/13
class	24/24	17/24	19/24	13/24	19/24	0/24	0/24	0/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	0/24	22/24	24/24	18/24	0/24	24/24
super	8/8	7/8	4/8	6/8	7/8	0/8	0/8	0/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	0/8	6/8	8/8	7/8	0/8	8/8
generators	27/27	24/27	24/27	16/27	0/27	0/27	0/27	0/27	27/27	27/27	27/27	25/27	27/27	27/27	27/27	27/27	27/27	27/27	27/27	27/27	27/27	27/27	27/27	0/27	16/27	27/27	0/27	20/27	27/27
Built-ins																													
typed arrays	46/46	0/46	45/46	0/46	45/46	0/46	8/46	16/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	46/46	18/46	37/46	46/46	46/46	43/46	46/46
Map	19/19	14/19	19/19	14/19	19/19	15/19	0/19	8/19	18/19	19/19	19/19	18/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	0/19	17/19	19/19	18/19	17/19	19/19
Set	19/19	14/19	19/19	14/19	19/19	15/19	0/19	8/19	18/19	19/19	19/19	18/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	19/19	0/19	18/19	19/19	18/19	17/19	19/19
WeakMap	12/12	6/12	12/12	9/12	12/12	0/12	0/12	6/12	11/12	12/12	12/12	11/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	0/12	9/12	11/12	11/12	11/12	12/12
WeakSet	11/11	5/11	11/11	8/11	11/11	0/11	0/11	0/11	10/11	11/11	11/11	10/11	11/11	11/11	11/11	11/11	11/11	11/11	11/11	11/11	11/11	11/11	11/11	0/11	9/11	10/11	10/11	10/11	11/11

http://kangax.github.io/compat-table/es6/

let

```
function calculateTotalAmount (vip) {
  var amount = 0 // probably should also be let, but you can mix var and let
  if (vip) {
    let amount = 1 // first amount is still 0
  }
  { // more crazy blocks!
    let amount = 100 // first amount is still 0
    {
      let amount = 1000 // first amount is still 0
      }
  }
  return amount
}

console.log(calculateTotalAmount(true))
```

https://webapplog.com/es6/

const

```
function calculateTotalAmount (vip) {
  const amount = 0
```

https://webapplog.com/es6/

Template strings

```
var name = 'Your name is ' + first + ' ' + last + '.'
var url = 'http://localhost:3000/api/messages/' + id
```

vs

```
var name = `Your name is ${first} ${last}.`
var url = `http://localhost:3000/api/messages/${id}`
```

https://webapplog.com/es6/

Fat Arrow Functions

```
var _this = this
$('.btn').click(function(event){
  _this.sendData()
})
```

vs

```
$('.btn').click((event) =>{
  this.sendData()
})
```

Autobinding!

https://webapplog.com/es6/

Classes

```
class baseModel {
  constructor(options = {}, data = []) { // class constructor
    this.name = 'Base'
    this.url = 'http://azat.co/api'
    this.data = data
    this.options = options
  }

    getName() { // class method
      console.log(`Class name: ${this.name}`)
    }
}
```

https://webapplog.com/es6/

Classes Inheritance

```
class AccountModel extends baseModel {
  constructor(options, data) {
```

```
let accounts = new AccountModel(5)
```

https://webapplog.com/es6/

Modules

module.js

```
module.exports = {
  port: 3000,
  getAccounts: function() {
    ...
  }
}
```

main.js

```
var service = require('module.js')
console.log(service.port) // 3000
```

https://webapplog.com/es6/

Many More Things

Generators

Spread operator

Promises

Symbols

For.. of loops

Map & Set

Tail calls

Put in next-gen JavaScript

```
[1, 2, 3].map(n => n ** 2);
```

Get browser-compatible JavaScript out

```
[1, 2, 3].map(function (n) {
  return Math.pow(n, 2);
});
```

Check out our REPL to experiment more with Babel!

Resources

http://es6-features.org/

https://github.com/lukehoban/es6features

http://exploringjs.com/es6/ch_overviews.html

https://hackernoon.com/javascript-es6-exploring-the-new-built-in-methods-b62583b0a8e6

https://scotch.io/bar-talk/five-things-you-can-use-in-es6-today