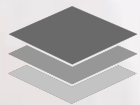




UI Engineering Studio. Day 8



# Bootcamp: JS ECMA 2015

## Ecma 2015

ECMAScript 2015 is the sixth edition of the ECMAScript Language Specification standard. It defines the standard for the [JavaScript](#) implementation in [SpiderMonkey](#), the engine used in Firefox and other Mozilla applications, Chrome's V8 and so on . . .

Const, Let. Destructuring

Symbol

Template Literals

Spread - Param handling

Arrow functions

## UI Boot Camp: JS Collections

# Const - Let

**Const** :you can only assign once a value in a variable. Only the reference is immutable, the value can change.

**Let**: The new var but better ;) New ability of the variable to be scoped to the block level.

Pen

```
let hey = [1, 2, 3, 4, 5];  
const name = 'Pipo';
```

## UI Boot Camp: JS Collections

# Destructuring

The **destructuring assignment** syntax is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables.

Pen

```
let [one, two] = [1, 2];
```

```
var a = 1
```

```
var b = 3
```

```
//Easiest way to swap the values ?
```

```
 //(a = 3 , b = 1)
```

## UI Boot Camp: JS Collections

# Symbol

New primitive !. Has the capability to be UNIQUE.

Pen

```
const name = Symbol('Pipo');
```

## UI Boot Camp: JS Collections

# Template literals

Template literals are string literals allowing embedded expressions. You can use multi-line strings and string interpolation features with them.

[Pen](#)

```
let a = 'better?'  
let b = 'strings'  
  
let subtitle = `this is a ${a} way to concat and handle ${b}`
```

## UI Boot Camp: JS Collections

# Spread and Rest

**Spread syntax** : allows an iterable such as an array expression or string to be expanded

**rest parameter** : syntax allows us to represent an indefinite number of arguments as an array.

Spread

Rest

```
//something to log stuff easier
let log = (...x) => { //this is actually a REST operator
  console.log(...x);
}

//create a simple array
let args = [0, 1, 2];
//log(...args);
```



## UI Boot Camp: JS Collections

# Arrow Functions

**Arrow function expression** has a shorter syntax than a function expression and does not have its own `this`, arguments or `super`. These function expressions are best suited for non-method functions, and they cannot be used as constructors.

[Pen](#)

```
var mainConciseSyntax = (x,y) => x+y;
```

## UI Boot Camp: HTML

# Homework: Sliding Puzzle

A combination puzzle that challenges a player to slide pieces along certain routes to establish a certain end-configuration. The pieces to be moved may consist of simple shapes, or they may be imprinted with colors, patterns, sections of a larger picture (like a jigsaw puzzle), numbers, or letters.

- [Wikipedia](#)



**Develop a Sliding Puzzle using HTML,CSS and JS.  
Please avoid the use of any library or framework.**

