ES6 [illegible]

[illegible] 10 [illegible]

[illegible] ECMAScript 6 [illegible].

let ו const [illegible]

Const [illegible] let. [illegible] var [illegible] const ו let [illegible] {} [illegible] const ו let [illegible].

```
function f() {
  var x = 1
  let y = 2
  const z = 3
  {
    var x = 100
    let y = 200
    const z = 300
    console.log('x in block scope is', x)
    console.log('y in block scope is', y)
    console.log('z in block scope is', z)
  }
  console.log('x outside of block scope is', x)
  console.log('y outside of block scope is', y)
  console.log('z outside of block scope is', z)
}

f()
```

const_let-example.js hosted with ♥ by GitHub

```
x in block scope is 100
y in block scope is 200
z in block scope is 300
x outside of block scope is 100
y outside of block scope is 2
z outside of block scope is 3
```

const_let-output.txt hosted with ♥ by GitHub

□□□□□□□□□□ □□□□ □□□□ □□□□□□□

□□□□□□ □□□□□□ □□□□□□□□□□□□ □□□□□ □□ □□□□□□□ □□□□□□□ □□□□□□□□ □□ □□□□□□ □□□□□ □□□□□ □□□□ ECMAScript5 □□ □□□□□□ □□□□□ □□□□□□□ □□□□□□□ □□□□□□ □□□□□ □□□□□ □□□□□□□ □□ □□ □□□□□ □□ □□□□□ □□□□ □□□□□ □□□□□ □□ □□□□ □□□□□. □□□□□ □□. □□□□□ □□□□□ □□□□□□□ □□□□□□□ □□□□□□ □□□□□ □□□□□ □□□□□□□ □□ □□ □□□□□ □□ □□□□□ □□□□ □□□□□ □□□□□ □□ □□□□ □□□□□. □□ □□. □□□□□ □□□□□ test □□□□□ □□□□□□□□ □□□□□□ □□ □□□□ .□□□□□□ □□□□□ □□ □□□□□□□ □□□□□□ □□□□□ □□□□ □□ □□□□□ □□ □□□□□□□□ □□□□□□□□ □□□□□□□ □□□□□□ □□□□□ □□ □□□□□□□ □□□□□□□ .□□□□□ □□□□□□ □□ □□□□□□□□ □□□□□□□□ □□□□□□□ □□□□□□ □□□□□ □□ □□□□□□□ □□□□□□□

```
function testFunction(x, y = 1, z = x + 1){
     return x / y + z;
};
let result = testFunction(1);
console.log(result); // 3
```

Spread □□□□□

□□ □□□□□□ □□□□□□ □□ □□ □□□□□ □□□□ □□□□□□ □□ □ □□□□□□ □□□□□ □□□□□ .... □□□□□ □□ □□□□□ □□ □□□□□□□ □□□□□□ □□ □□□□□□ □□□□ □□□□□□□□□□ □□□□□ testFunction □□□□□ .□□□□□□□ □□□□□ □□□□□□□□ □□□□□ □□□□ □□ □□□□□□ □□□□□□□ □□□□□ □□ □□□□□□.

```
let testFunction = (x, y) => x + y;
let data = [2,3];
let result = testFunction(...data);
console.log(result); // 5
```

.□□□□ □□□□□□ □□□□□□ □□□□□ □□□□□ □□ □□ □□□□□□ □□ □□□□□□□ □□□□□□□□□ □□□□□□ □□□□□ □□ □□□□□□□□ □□

```
let firstArray = [3,4];
let secondArray = [1,2,...firstArray, 5];
console.log(secondArray); // Array [1, 2, 3, 4, 5]
```

Array destructuring assignment

□□□□□□□□□ □□ □□□□□□□ □□ □□□□□□□ □ □□□□□□ □□□□□□ □□ □□ □□□□□ □□ □□□□□□□ □□□□□□□□ □□□□□□ □□ □□□□□□□ □□□□ □□□□□□ □□ □□□□□□□□□ ECMAScript6 □□ □□□□□ □□□□□□ .□□□□□□□ □□□□□ □□□□□□□□ □□□□□ □□□□ □□ □□□□□□ :□□□□□ □□□□□□□□

```
let testArray = [1, 2, 3];
let x, y, z;
[x, y, z] = testArray;
console.log(x, y, z); // 1 2 3
```

□□□□□□□□ □□□□□□ □□□□ □□□□□ □□□□□□ □□ □□ □□□□□ □□□□□□ □□ □□□□□□□□□ z □ x□ y □□ □□□□□□□ □□□□□□□ □□□□ □□ □□□□ □□□□□ □□□□□ □□ □□□□□□ .□□□□

```
let testArray = [1, 2, 3];
let x, y, z;
x = testArray[0];
y = testArray[1];
z = testArray[2];
console.log(x, y, z); // 1 2 3
```

□□□□□ □□□□□□ □□□□□ □□ □□□□□□□ □□□□□□ □□□□ □□ □□ □□□□□□□□□□ □□ □□□□ □□□□ □□□□□ □□□□□□ □□□□□ □□ □□□□□ □□ □□□□□ □□□□ □□ □□□□ .□□□□□ □□□□□ □□□□□□ □□□□□□ □□□□□□ □□ □□□□ □□ □□ □□□□

□□□□□□□

□□□□□□ □□ □□□□□□□ □□□□□ □□□□□ □□ □□□□□□□ □□□□□□□□□□□□□ □□□□□□□□□ □□□□□ □□ □□ □□□□□□ □□□□□ □□□□□□□□□□□□□□□ □□□□ □□□□□□□□□□□□□ □□ □□□□□□□ □□ □□□□□ □□□□□□□ □□□□□□ □□ □□□□□□ □□□□□ □□ □□□□□□□□□□□□□□□ □□. □□□□□□□□ □□□□□ □□□□ □□ .□□□□□ □□□□□□□ ES6 □□ □□□□□□ □□□□□□□ □□ □□ □□□□□ □□□□□□□□□□□□□□□ □□□□□□ □□□□ □□□□□ □□□□□□□□□ .□□ □□□□□□ □□□□□□□□□□□□□□□ □□□□□□□ □□□□□ □□□□□ □□□□□□ □□□□□ □□□□□□□ □□□□□ □□ □□□□□□□ □□□□□□ □□ □□□□□□□

```
class Point {
    constructor(x, y) {
        this.x = x
        this.y = y
    }

    toString() {
        return '[X=' + this.x + ', Y=' + this.y + ']'
    }
}

class ColorPoint extends Point {
    static default() {
        return new ColorPoint(0, 0, 'black')
    }

    constructor(x, y, color) {
        super(x, y)
        this.color = color
    }

    toString() {
        return '[X=' + this.x + ', Y=' + this.y + ', color=' + this.color + ']'
    }
}

console.log('The first point is ' + new Point(2, 10))
console.log('The second point is ' + new ColorPoint(2, 10, 'green'))
console.log('The default color point is ' + ColorPoint.default())
```

Find

[illegible] true [illegible] false [illegible].

```
var people = [
  {name: 'Jack', age: 50},
  {name: 'Michael', age: 9},
  {name: 'John', age: 40},
  {name: 'Ann', age: 19},
  {name: 'Elisabeth', age: 16}
]

function teenager(person) {
    return person.age > 10 && person.age < 20
}

var firstTeenager = people.find(teenager)

console.log('First found teenager:', firstTeenager.name)
```

Arrow functions

.□□□□ □□□□ □□□□□ □□□□□□ □□□□□□□□□□ □□ product □□ sum □□□□□ □□□□□□ □□□□□ □□□□□□□□□□ □□ □□□□ □□□□.
.□□□□□□□ □□□□□□□□ □□ □□□□ □□□□ □□□□□□ .□□□□ □□□□□□□□ Arrow Function □□□□□□ □□ □□□□□□□□□ □□□□ □□□□

```
var array = [1, 2, 3, 4]

const sum = (acc, value) => acc + value
const product = (acc, value) => acc * value

var sumOfArrayElements = array.reduce(sum, 0)
var productOfArrayElements = array.reduce(product, 1)
```

.□□□□□□ □□□□ □□ □□□ □□□□□□□□ □□□□ □□ □□□□□□□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□□□□ □□□□□□ □□□□ □□ □□□□□□ □□□□□ □□□□□□□□ □□□□ □□ □□□□ □□ □□□□□□□□

```
var array = [1, 2, 3, 4]

var sumOfArrayElements = array.reduce((acc, value) => acc + value, 0)
var productOfArrayElements = array.reduce((acc, value) => acc * value, 1)
```

.□□□ □□□□□ □□□□□□□□ □□□□□ □□□□ □□□□□□□□□□ □□□□ □□ □□□□□□□ □□ Arrow functions

```
var array = [1, 2, 3, 4]

const sum = (acc, value) => {
  const result = acc + value
  console.log(acc, ' plus ', value, ' is ', result)
  return result
}

var sumOfArrayElements = array.reduce(sum, 0)
```

[illegible]

[illegible] ES6 [illegible] placeholder[illegible]

```
function hello(firstName, lastName) {
  return `Good morning ${firstName} ${lastName}!
How are you?`
}

console.log(hello('Jan', 'Kowalski'))
```

Promise

Promise [illegible] then [illegible] catch [illegible]

```
function asyncFunc() {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
          const result = Math.random();
          result > 0.5 ? resolve(result) : reject('Oppps....I cannot calculate')
        }, 1)
    });
}

for (let i=0; i<10; i++) {
        asyncFunc()
        .then(result => console.log('Result is: ' + result))
        .catch(result => console.log('Error: ' + result))
}
```

:□□□□□□□
□□□□□ □□□□□□□
:□□□□ □□□□
□□□□□ □□□□□
□□□□□ □□□□□□
:□□□□□□ □□□□□
16:00 - 30/07/1396
:□□□□□
- □□□□□□□□□□□□ □□ □□□□□□□□□□□□ - □□ □□□□□□□□□□□ - □□ □□□□□□□□□□□□ - □□□□□□□□□□□□ - ECMAScript 6
javascript - □□□□□□□□ □□□□ □□□□□□□□ □□□□□□ □□□□ □□□□□□ 10 - □□□□□□□□□□□□ □□□□□□□□□

□□□□□
□□□□:https://www.shabakeh-mag.com/news/world/10278/10-%D9%88%DB%8C%DA%98%DA%AF%D
B%8C-%D8%AC%D8%AF%DB%8C%D8%AF-%D9%88-%D8%AC%D8%B0%D8%A7%D8%A8-
%D8%AC%D8%A7%D9%88%D8%A7%D8%A7%D8%B3%DA%A9%D8%B1%DB%8C%D9%BE%D8%A
A-%D8%A8%D8%B1%D8%A7%DB%8C-%D8%B9%D8%A7%D8%B4%D9%82%D8%A7%D9%86-
%D9%88%D8%A8