

JavaScript

/

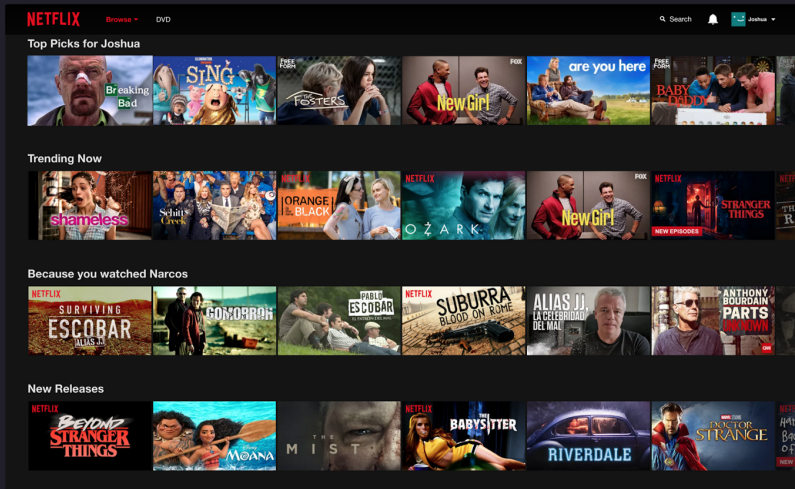
ECMAScript

Made

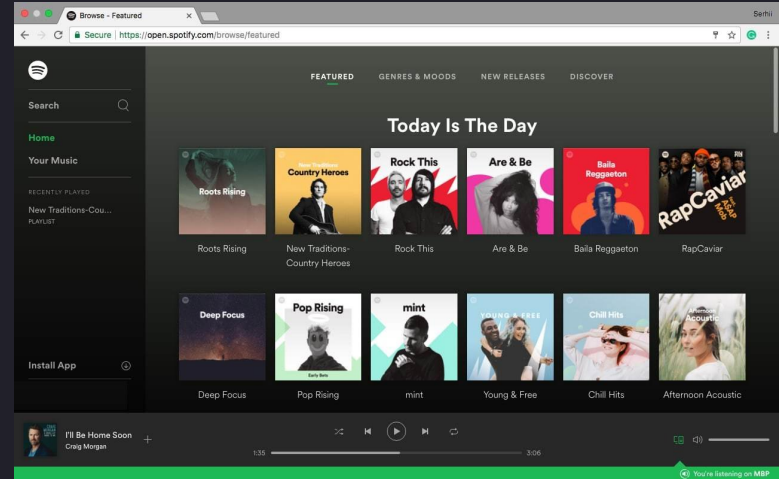
1995



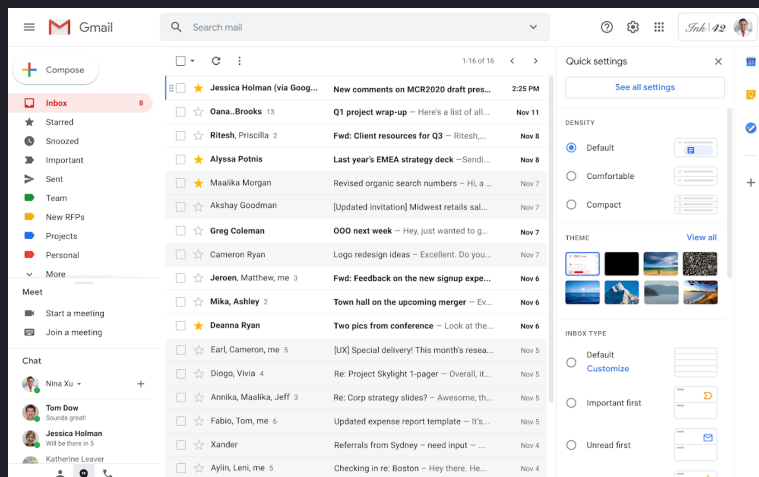
Powers the web



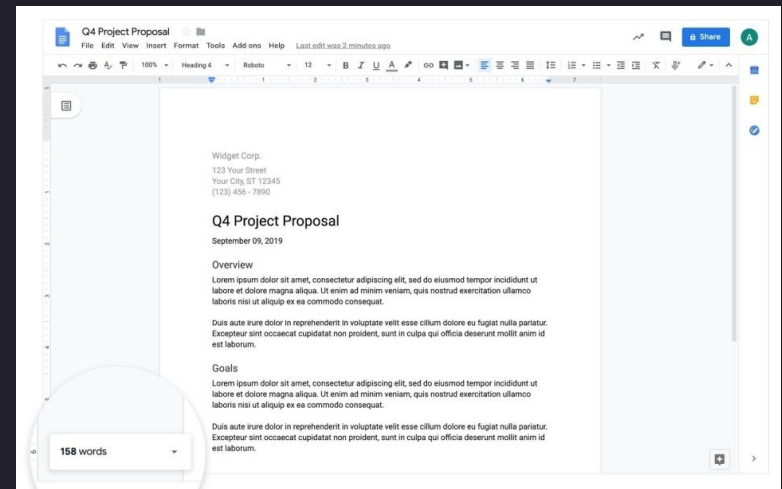
Netflix



Spotify



Gmail



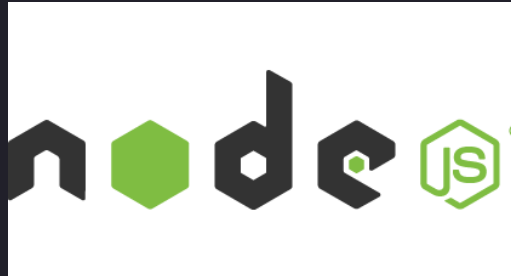
Google Docs

JavaScript Today

Browser:



Backend:



Hardware:



+++

Reference

MDN Docs

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/>

Variables

```
var a = "Hallo";  
var b;  
var c = 5;  
d = 10;
```

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables

Variables

```
let a = "Hallo";  
let b;  
let c = 5;  
d = 10;
```

Re-assignment

```
let a = "Hallo";  
a = "Goodbye";
```


Constants

```
const program = "Academy";  
const weeks = 12;
```

```
// Cannot be re-assigned
```

Dynamically typed

```
let a = "Hallo";  
a = 15;  
a = true;
```

Scope

Blocks

```
if (true) {  
    const a = 5;  
}
```

```
console.log(a);
```

```
// => Uncaught ReferenceError:  
//    a is not defined
```

Scope

Functions

```
function yelp() {  
    var a = 5;  
}
```

```
yelp();
```

```
console.log(a);  
// => undefined
```

Scope

Global

```
function yelp() {  
    a = 5;  
}
```

```
yelp();
```

```
console.log(a);
```

```
// => 5
```

Hoisting

```
console.log(a);
```

```
var a = 5;
```

```
console.log(a);
```

Hoisting

```
var a;  
  
console.log(a);  
  
a = 5;  
  
console.log(a);
```

Variable declarations are hoisted to the top of the scope!

Functions

```
function add(a, b) {  
    return a + b;  
}
```

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions>

Functions

```
function add(a, b) {  
    return a + b;  
}
```

```
let result = add(10, 20);  
console.log(result) // 30
```

More in depth later

If statements

```
if (a < 5) {  
    console.log(a, "less than 5");  
}  
  
if (name == "Eirik") {  
    console.log("Name is Eirik");  
} else if (!name) {  
    console.log("No name!");  
} else {  
    console.log("Name set, but it ain't Eirik");  
}
```

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else>

If statements

Optional brackets for one-liners

```
if (isWinner)  
    showWinnerScreen();
```

```
if (myCoolFunction)  
    myCoolFunction();
```

```
if (!result)  
    return;
```

More about equality later

switch statements

```
var choice = 3;

switch (choice) {
  case 1:
    console.log("You chose 1, yay!");
    break;
  case 2:
    console.log("You chose 2 :-)");
    break;
  default:
    console.log("You chose something else..");
    break;
}
```

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/switch>

switch statements

```
var direction = "up";

switch (direction) {
  case "up":
    console.log("You're moving up!");
    break;
  case "down":
    console.log("You're moving down!");
    break;
  default:
    console.log("I'm not sure where you're going..");
    break;
}
```

switch on mixed types

```
switch (choice) {  
  case 1: {  
    console.log("Hey number 1");  
    break;  
  }  
  case "Oops": {  
    console.log("String");  
    break;  
  }  
  case 1 + 2: {  
    console.log("Even expressions");  
  }  
}
```

While loop

```
var i = 0;

while (i < 10) {
  console.log(i);
  i += 1;
}
```

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while>

for Loops

```
for (var i = 0; i < 10; ++i) {  
    console.log(i);  
}
```

```
var i;  
for (i = 0; i < 10; ++i) {  
    console.log(i);  
}
```

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for>

Break

```
for (var i = 0; i < 10; ++i) {  
    if (i == 5) {  
        break;  
    }  
  
    console.log(i);  
}
```

Continue

```
for (var i = 0; i < 10; ++i) {  
    if (i == 5) {  
        continue;  
    }  
  
    console.log(i);  
}
```

Exceptions

```
try {  
    // try some code here  
} catch (e) {  
    // handle errors here  
} finally {  
    // optionally - always do this  
}
```

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/try...catch>

https://developer.mozilla.org/en-US/docs/web/javascript/reference/global_objects/error

Exceptions

```
try {  
    // try some code here  
} catch (e) {  
    // handle errors here  
} finally {  
    // always do this (optional)  
}
```

Exceptions

```
try {  
    throw new Error("Hey, don't do that!");  
} catch (e) {  
    console.log(e.name, e.message);  
    // "Error", "Hey, don't do that!"  
} finally {  
    console.log("Always run this");  
}
```

Throw anything

```
try {  
    throw "Don't do that";  
} catch (e) {  
    console.log(e);  
    // "Don't do that"  
}
```

Automatic Semicolon Insertion

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#automatic_semicolon_insertion

ASI

What is the return value?

```
function returnIt() {  
    return  
    5 + 5  
}
```


ASI

Not what you would expect

```
function returnIt() {  
    return;  
    5 + 5;  
}
```

ASI

What is the result?

```
var b = 1, c = 2, d = 3, e = 4;
```

```
var a = b + c  
(d + e).toString()
```

Oh no!

```
TypeError: number is not a function
```

Interpreted as

```
var b = 1, c = 2, d = 3, e = 4;
```

```
var a = b + c(d + e).toString();
```

Solution

```
var b = 1, c = 2, d = 3, e = 4;
```

```
var a = b + c;  
(d + e).toString();
```

Tasks

Canvas: [exercises-basic-syntax.zip](#)