

The image features the JavaScript logo, which consists of the letters 'JS' in a bold, black, sans-serif font. The letters are centered within a bright yellow square. This yellow square is itself centered within a larger, solid gray rectangle that serves as the background for the entire image.

JS

From humble beginnings...

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) ,[Mail robot](#) ,[Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

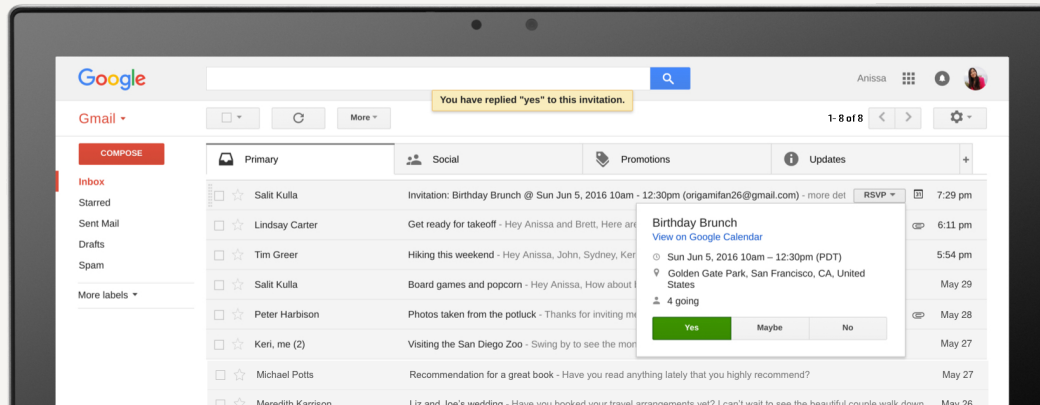
[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

...to Gmail...



...to...THE FUTURE



(no really: WebVR)



where it began

nineteen-ninety what?



I like Eich



I like Eich



Running JavaScript

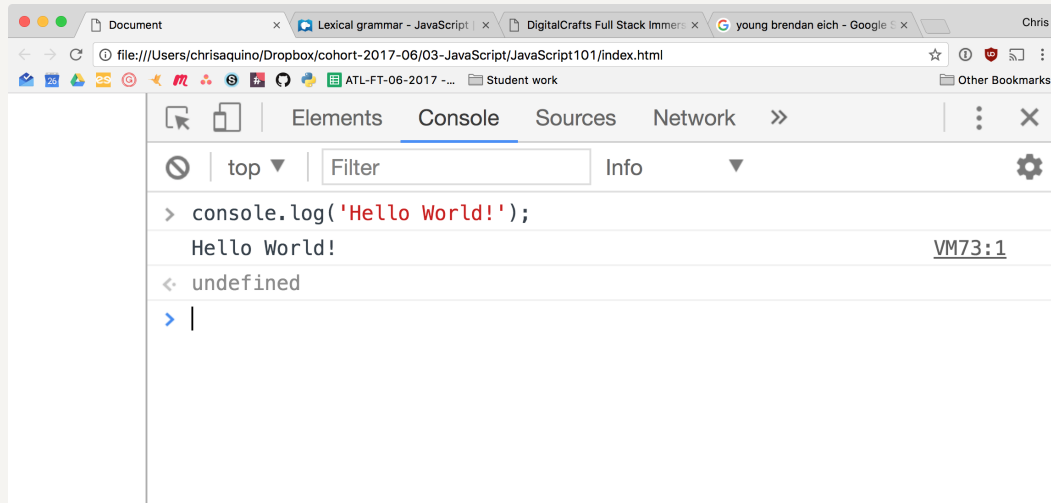
Python Tutor

The screenshot shows the Python Tutor web application in a browser. The browser's address bar displays the URL `pythontutor.com/visualize.html#mode=edit`. The page title is "Visualize Python, Java, JavaScript". On the left side, there is a button labeled "Start shared session" and a link "What are shared sessions?". The main area is a code editor titled "Write code in" with a dropdown menu set to "JavaScript ES6". The code editor contains the following code:

```
1 // Look Ma, no Python!
2
3 var salutation = 'Hello Builders!';
4 alert(salutation);
```

Support our research and keep this tool free by filling out this [survey on how your native spoken language affects how you learn programming](#).

Chrome Dev Tools



Create a new project

```
mkdir JavaScript101  
cd JavaScript101  
touch index.html  
mkdir scripts  
touch scripts/main.js
```

Create an HTML skeleton in index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <script src="scripts/main.js"></script>
</body>
</html>
```

With a script tag!

Add 'Hello World' to main.js

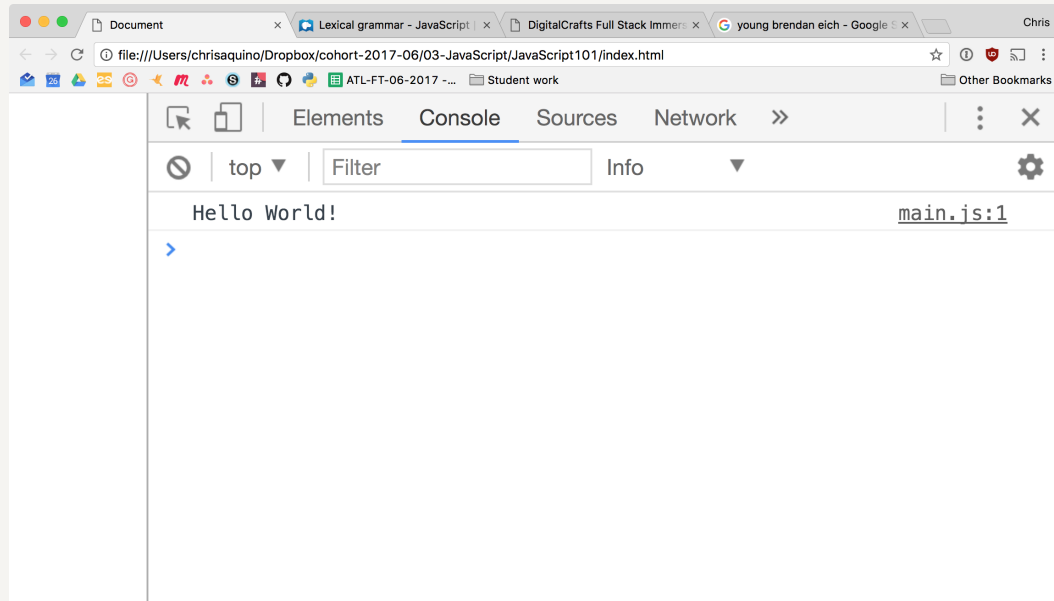
```
console.log('Hello World!');
```

Open index.html in Chrome

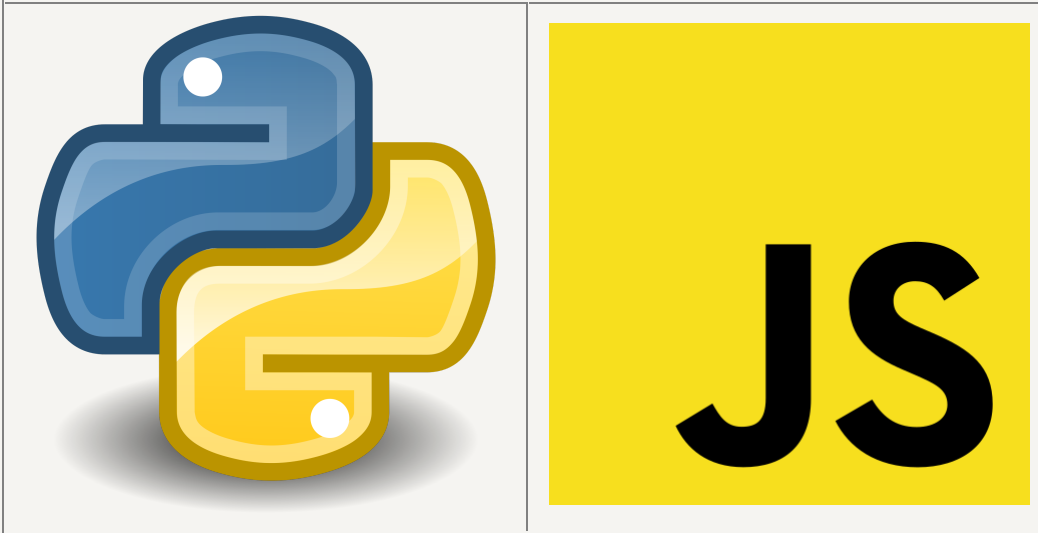
```
open index.html
```

Or just double click the file index.html

Open the Dev Tools



Comparison overview



Printing

Printing

python

```
print 'Hello World'
```

Printing

python

```
print 'Hello World'
```

JavaScript

```
console.log('Hello World');
```

Strings

Strings

python

```
"Hello!"  
'Hello!'  
"""Hello  
There!  
"""
```

Strings

python

```
"Hello!"  
'Hello!'  
"""Hello  
There!  
"""
```

JavaScript

```
"Hello!"  
'Hello!'  
`Hello  
There!  
`
```


Concatenation is the same

Concatenation is the same

python

```
"Hello" + " " + "There"
```

Concatenation is the same

python

```
"Hello" + " " + "There"
```

JavaScript

```
"Hello" + " " + "There"
```

Variables

- JS *needs* the `var` keyword
- otherwise, bad things happen...
 - a global variable is created, even from inside a function

Variable names

- can start with a-z, A-Z, _, \$ (yes, dollar)
- after first letter, 0-9, a-z, A-Z, _, \$
- can't use **reserved words**

Use camelCase, not snake_case

```
var veryLongVariableName = "";  
// Yes!  
  
var very_long_variable_name = "";  
// Nope.
```

String length

- No len function
- Uses the .length property

```
var hey = "Hello There";  
hey.length  
// 11
```

String indexing: two ways

```
var hey = "Hello There";  
hey[1]  
// e  
  
hey.charAt(1)  
// e
```


Numbers: all floats, all the time!

```
3.0  
// 3  
  
3  
// 3  
// ^^ actually a float
```

Division

8/3

// 2.666666

Converting Strings to Numbers

```
Number('5');  
// 5
```

Converting Strings to Numbers

```
Number('5');  
// 5
```

Alternatively: parseInt

```
parseInt('5', 10); // Specify that we're in base-10  
// 5
```

NaN

NaN

python

```
int('blah')
```

NaN

python

```
int('blah')
```

JavaScript:

```
Number('blah')
```

Converting Number to String

Converting Number to String

python

```
str(45)
```

Converting Number to String

python

```
str(45)
```

JavaScript

```
String(45)
```

Math operations

- arithmetic is the same
- certain math operations in JS require a prefix:
 - `Math.abs`
 - `Math.pow`
 - `Math.round`
- no need to import (thanks, browser!)

User Input

User Input

python

```
name = raw_input('What is your name? ')
```

User Input

python

```
name = raw_input('What is your name? ')
```

JavaScript

```
var name = prompt('What is your name? ')
```

if statements

like python, you have three parts:

- `if` keyword
- a condition (but *must* be wrapped in parens)
- a body (but *must* be wrapped in curly braces)

python

```
if age >= 21:  
    print 'you can booze'  
elif age >= 16:  
    print 'you can car'  
else:  
    print 'you can run a startup'
```


JavaScript

```
if (age >= 21) {  
  console.log('you can booze');  
} else if (age >= 16) {  
  console.log('you can car');  
} else {  
  console.log('you can run a startup');  
}
```

More about booleans

More about booleans

logical and

and

&&

More about booleans

logical or

```
or
```

```
||
```

Functions

A function

```
function greet() {  
  console.log('hello builders!');  
}  
greet();
```

A function, with an argument

```
function greet(whom) {  
  console.log('hello ' + whom + '!');  
}  
greet('builders');
```

A function, with an argument

```
function greet(whom) {  
  console.log('hello ' + whom + '!');  
}  
greet('builders');
```

Same, but with a template string

```
function greet(whom) {  
  console.log(`hello ${whom}!`);  
}  
greet('builders');
```


A function, with multiple arguments

```
function greet(whom, myName) {  
  console.log(`hello ${whom}! I am ${myName}.`);  
}  
greet('builders', 'the great and terrible Oz');
```

A function, returning a value

```
function greet(whom, myName) {  
  return `hello ${whom}! I am ${myName}.`;  
}  
var message = greet('builders', 'the great and terrible Oz');  
console.log(message);
```

A function, with a variable

```
function greet(whom, myName) {  
  var msg = '';  
  msg = `hello ${whom}! I am ${myName}.`;   
  return msg;  
}  
var message = greet('builders', 'the great and terrible Oz');  
console.log(message);
```

A function, argument left off

```
function greet(whom, myName) {  
  var msg = '';  
  msg = `hello ${whom}! I am ${myName}.`;   
  return msg;  
}  
var message = greet('builders');  
console.log(message);
```

A function, checking for argument values

```
function greet(whom, myName) {  
  var msg = '';  
  if (myName) {  
    msg = `hello ${whom}! I am ${myName}.`;   
  } else {  
    msg = `hello ${whom}!`;   
  }  
  
  return msg;  
}  
var message = greet('builders');  
console.log(message);
```

A function, providing a default using ||

```
function greet(whom, myName) {  
  var msg = '';  
  myName = myName || 'me';  
  msg = `hello ${whom}! I am ${myName}.`;   
  return msg;  
  
  return msg;  
}  
var message = greet('builders');  
console.log(message);
```

A function, providing a default argument value

```
function greet(whom, myName='me') {  
  var msg = `hello ${whom}! I am ${myName}.`;   
  return msg;  
}  
var message = greet('builders');  
console.log(message);
```

that's it...for now.