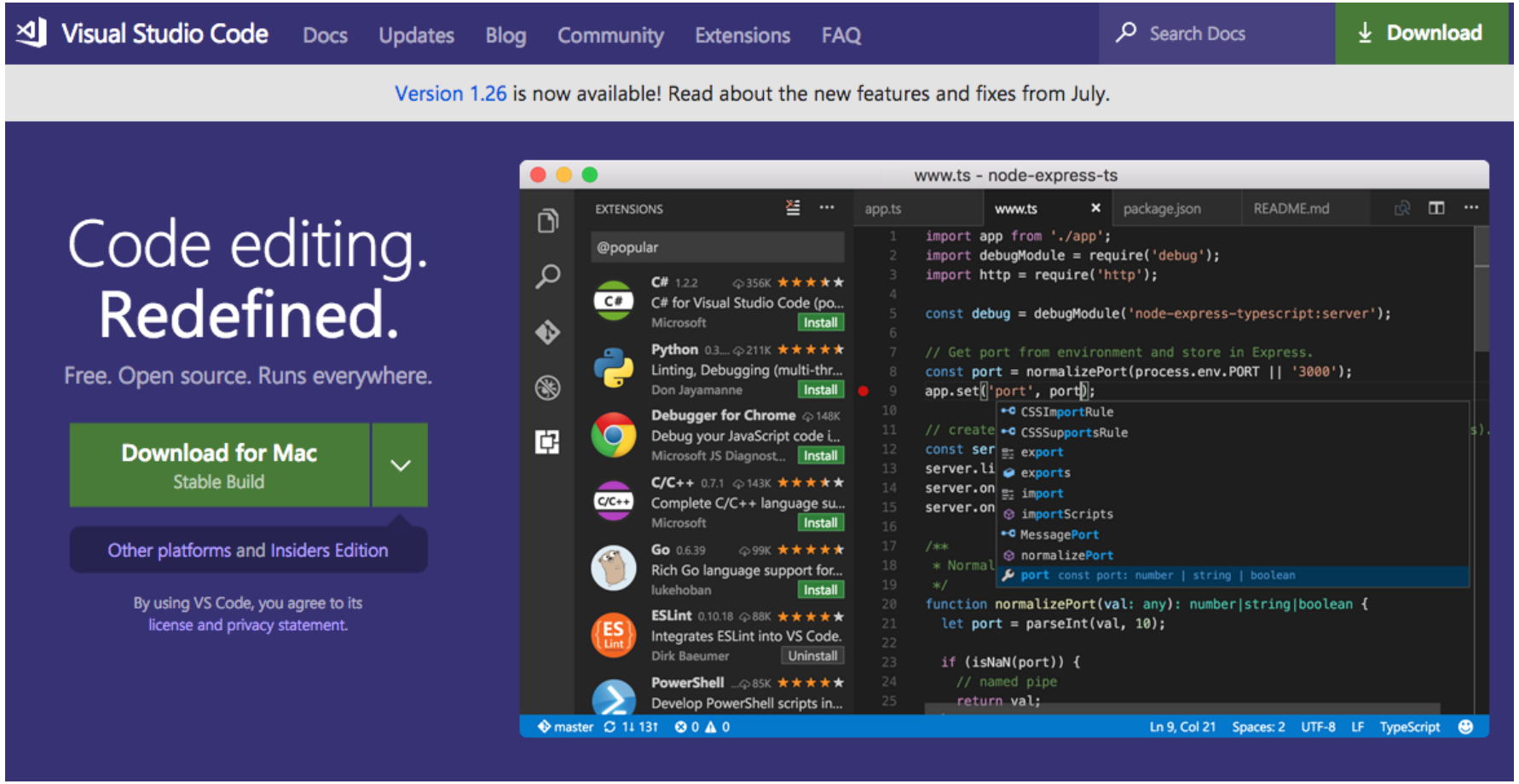
JavaScript

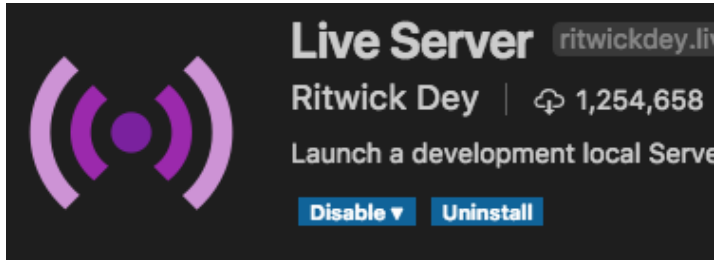
Index

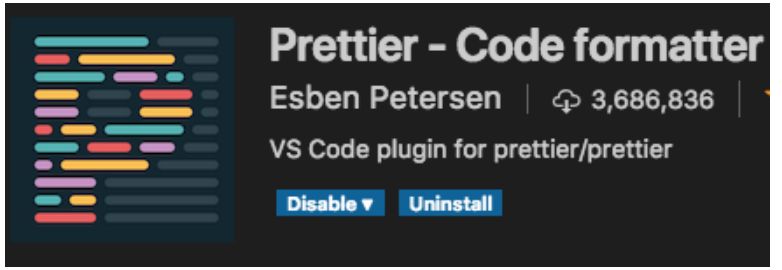
Environment Setup

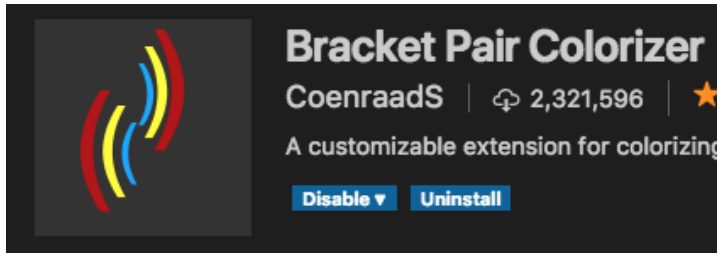
Setup – Visual Studio Code

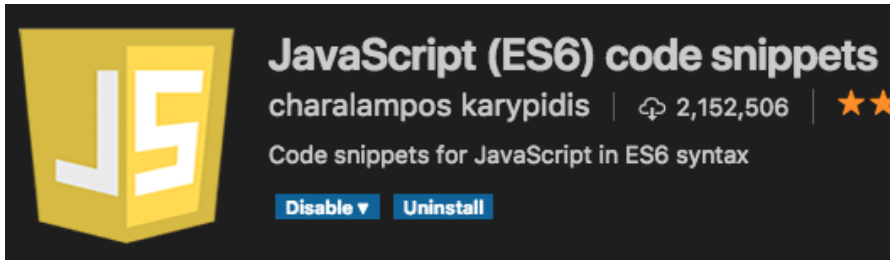


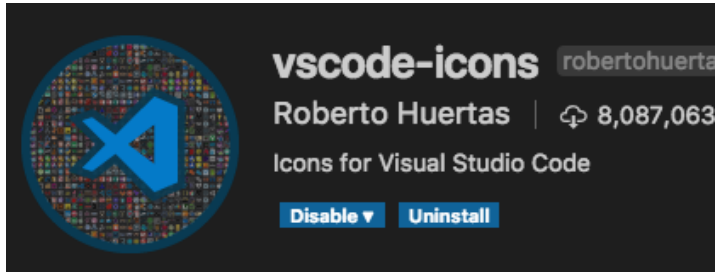
Setup – Visual Studio Code extensions











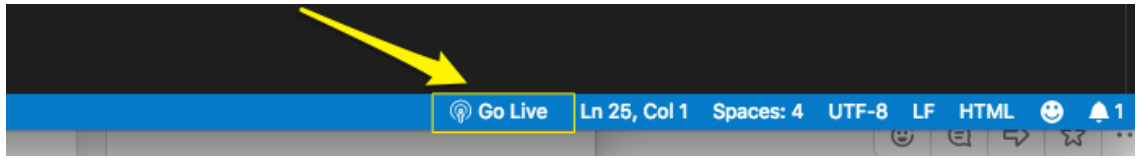
User Settings

```
"editor.formatOnSave": true,
"html.format.indentHandlebars": true,
"html.format.indentInnerHtml": true,
```

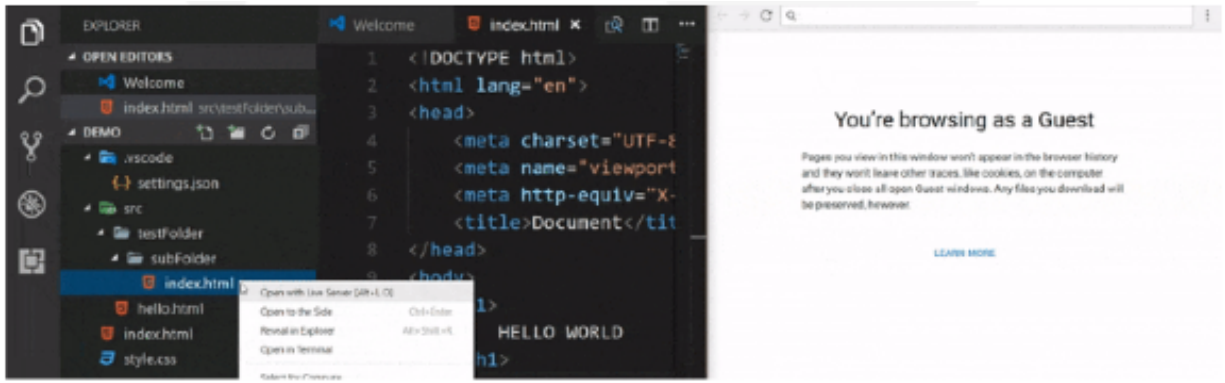
Live Server Extension

- Launch a development local Server with live reload feature

1. Open a project and directly click to Go Live from StatusBar to turn on/off the server



2. Rigth click on a HTML file and click “Open with Live Server”



Symbols

Chapter 1: Language Fundamentals

JavaScript

But, what is JavaScript after all?

Let the games begin: jump in!

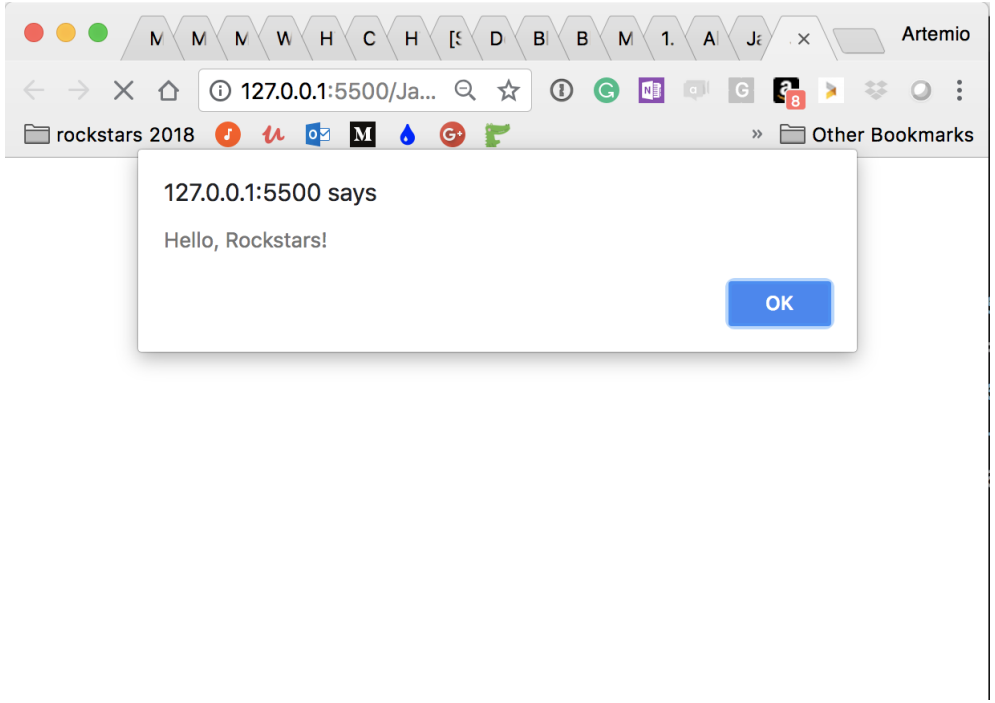
```html
<!DOCTYPE html>
<html lang="en">

    <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>JavaScript First Dive!</title>
    </head>

    <body>
        <script>
            alert('Hello, Rockstars!')
        </script>

    </body>

</html>
```

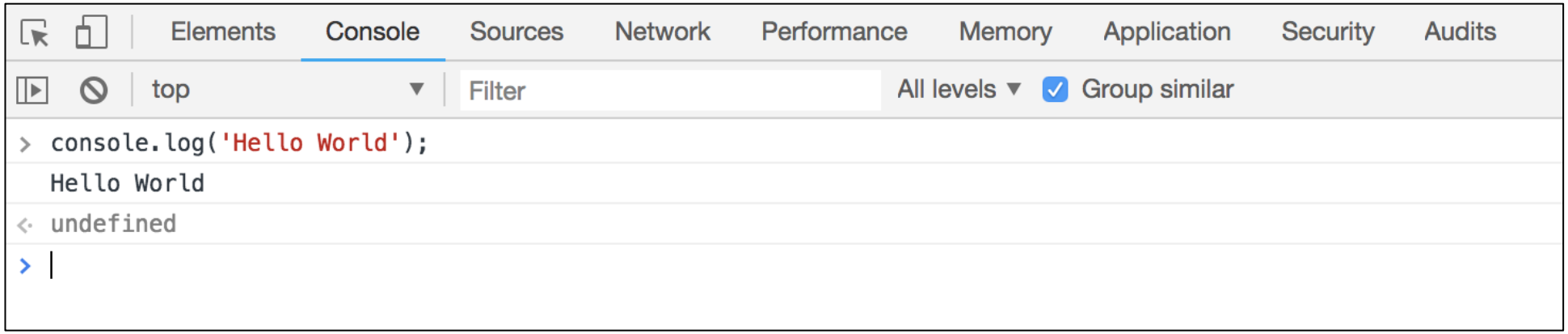


Built in web development tools: the console

- The five most popular web browsers (safari, IE, Edge, Chrome, Firefox, Opera) have built in tools.
- The elements allows to navigate the HTML DOM
- Web development tools commonly include a panel to debug scripts by allowing developers to add watch expressions, breakpoints, view the call stack, and pause, step over, step into, and step out of functions while debugging JavaScript
- The console is a common included tool, allowing to type in JavaScript commands and call functions, or view errors.
- The console is frequently used as output for debugging. This is the web development tool that will be used most during this training.

How to start the console

- **Google Chrome:** Ctrl + Shift + J (Win/Linx) or Cmd + Opt + J (Mac)
- **MS Edge or IE:** F12 to Developer Tools. Then navigate to Console Tab
- **Firefox:** Ctrl + Shift + K (Win/Linux) or Ctrl + Opt + K (Mac)



Variables in JavaScript

- Variables can ONLY start with letter, underscore or a dollar sign
- JavaScript is case sensitive
- **Strongly typed languages**: must declare datatypes (c/c++, java, c#)
- **Loosely typed languages**: not required to declare datatatypes
- JavaScript is a *loosely typed* language (or *dynamic*).
- The datatype of a variable is assigned when the variable receives a value
- There are supersets of JS and addons to allow static typing (TypeScript, Flow)

Variables declaration

- Variables can be declared by any of these keywords *var, let* and *const*:
  - ***var*** *variableName*
  - ***let*** *variableName*
  - ***const*** *variableName*
- ***var*** defines a variable global or locally to an entire functions, creating issues with block-level scoping
- var is an old feature (ES5), only included mostly for backwards compatibility. DO NOT USE!
- Since ES6 (EMACS 6) **let** and **const** were introduced to fix **var**
- ***let*** allows to modify initial value
- ***const*** does not allow to change value for *primitives*.
- Always use ***const***, unless you need to modify initial value of the variable

Variables – primitive datatypes

- Primitive datatypes: immutable, simple values, non-objects:
- JavaScript provides six primitive datatypes
  - **String** – sequence of text. Enclosed in “ or ‘ marks
  - **Number** – A number. Do not need quote marks around them
  - **Boolean** – a True or False value. The words *true* and *false* are reserved words
  - **Null** – always one value: ***null***.
  - **Undefined** – a variable without value has an ***undefined*** value
  - **Symbol** (introduced in ECMAS 6)
- The operator ***typeof*** retrieves the variable type.
- Further info: https://www.vojtechruzicka.com/javascript-primitives

Variables – no primitive datatypes

- Non Primitive DataTypes: objects with values *and* properties. They are mutable (can change)
  - Arrays – Structure that allows to store multiple values
  - Object Literals
  - Functions
  - Dates
  - Sets (ECMAS6)
  - Maps (ECMAS6)
  - Anything Else
- A variable can be used *before* being declared (?)

Variables - conventions

- Use **camelCase** convention for variable names
- Use character $ at the beginning of a variable name only to assign Jquery objects
- Use character _ at the beginning of a variable name only for patterns and advance models
- Use upper case character at the beginning of a variable name only for Objects (classes, constructors)
-  **const** does not define a constant value, but a constant reference to a value. Hence, a primitive value cannot change, but properties of a constant object can change

Variables – Object Wrappers

- What are the expected results for:
  - typeof new number(4);
  - typeof 4;
  - typeof new string('String');
  - typeof 'String';
  - typeof new Boolean(true);
  - typeof true
- What is the result of *null typeof* and why?
  - typeof null

Variables – Datatype conversion

- Convert *variable* To String:
  - String(*variable*)
  - *variable*.toString()

- Convert *variable* To Number:
  - Number(*variable)*
  - parseInt(*variable)*
  - parseFloat(*variable)*

- Not-a-Number or NaN, is a returned as error. It is an object.

Variables in JavaScript

- Create a JavaScript Sandbox
  - index.html
  - app.js
  - link app.js to index.html

Operators and The Math Object

- Artihmetic operators:
  - Addition(+)
  - Subtraction(-)
  - Division(/)
  - Multiplication(*)
  - Remainder(%)
  - Exponentiation(**)
  - Increment(++)
  - Decrement(--)
  - Unary negation(-)
  - Unary plus(+)

- The Math object *Math provides properties and methods for mathematical constants and functions:*
  - *Math.pi*
  - *Math.random()*
  - *Math.abs()*
  - *Math.round()*
  - *Math.ceil()*
  - *Math.floor()*
  - *Math.sin()*
  - *Math.sqrt()*
  - *Math.pow()*

String Methods and Concatenation

- Concatenation
  string1 + string2 + string3
- Append
  string1 += string2
- Method concat
  - *string.concat(string1,string2)*
- Escaping
  For keywords, use scape character **
  'This is a \'scape'\ example'
- Length
  - *string.length*

- Change Case (wrapper)
  string.toUpperCase();
  string.toLowerCase();
- Strings as arrays
  variable = 'I rock!'
  variable[0]
  variable[3]
- Search character
  string.indexOf('char');
  string.lastIndexOf('char');
  string.charAt('index')