

JS

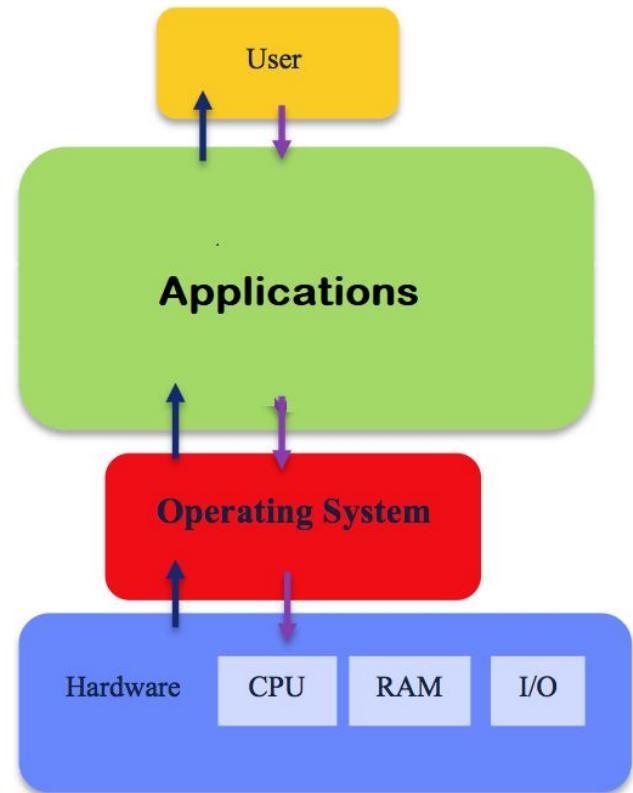




How does a computer work ?



Operating System





What Is Computer Program ?

なにをしたい
ですか？

What do you want to do?



おも
しろ
ほん
面白い本を
よ
読みたいです

I want to read a good book



o:\ Select Command Prompt

Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\zoela>help!
'help!' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\zoela>what now??????!!!
'what' is not recognized as an internal or external command,
operable program or batch file.

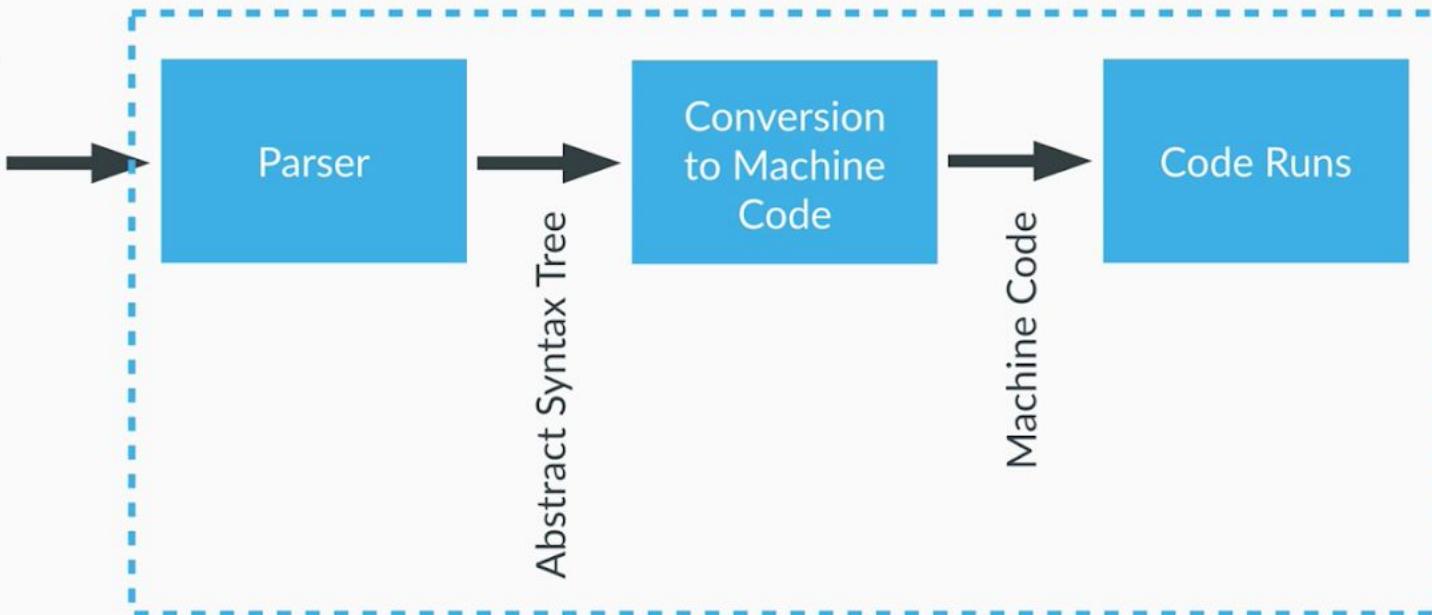
C:\Users\zoela>



OUR CODE

```
function calculateAge(yearOfBirth) {  
    return 2016 - yearOfBirth;  
}  
  
var johnsAge = calculateAge(1990);  
  
function yearsUntilRetirement(name, yearOfBirth) {  
    var age = calculateAge(yearOfBirth);  
    var retirement = 65 - age;  
    if (retirement >= 0) {  
        console.log(name + ' retires in ' + retirement + ' years.');//  
    } else {  
        console.log(name + ' is already retired.');//  
    }  
}  
  
yearsUntilRetirement('John', 1990);
```

JAVASCRIPT ENGINE





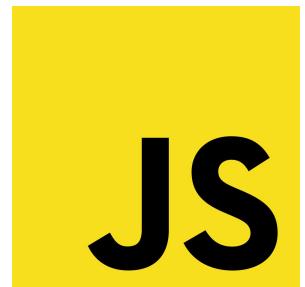
Quiz

Slido Link :

<https://wall.sli.do/event/pzze8630?section=e91127fe-c481-459a-94bd-b34f0c4f28fe>

Javascript

- Javascript is a scripting language that is one of the core to develop websites.



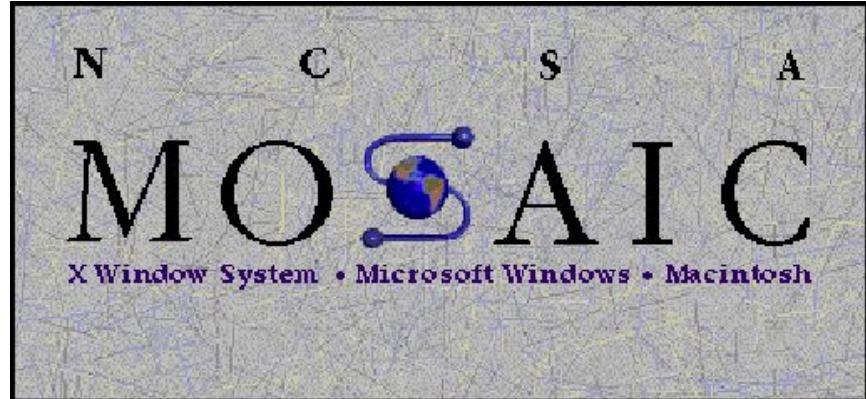
THE HISTORY OF JAVASCRIPT

`</>`

Let's go back to basics.
How much do you know
about the history of the most
popular programming
language?

Story of Javascript

- In 1994, Developer name Marc Andreesen published a web browser called Mosaic Navigator.



Netscape Navigator

- Later , Its name was changed to Netscape Navigator.



File Edit View Go Bookmarks Options Directory Help

Netscape: Version 1.1N

Back Forward Home Reload Images Open Print Find Stop N

Location: about:

What's New? What's Cool? Handbook Net Search Net Directory Newsgroups

 **Netscape Navigator™**
Version 1.1N
Copyright © 1994-1995 Netscape Communications Corporation, All rights reserved.

This software is subject to the license agreement set forth in the [License](#). Please read and agree to all terms before using this software.
Report any problems through the [feedback page](#).

NETSCAPE

 RSA™
PUBLIC KEY CRYPTO SYSTEM
Contains security software from RSA Data Security, Inc.
Copyright © 1994 RSA Data Security, Inc. All rights reserved.
This version supports International security with RSA Public Key Cryptography, MD2, MD5, RC4.

Any provision of Netscape Software to the U.S. Government is with "Restricted rights" as follows: Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contracting and manufacturing is Netscape Communications Corporation, 501 East Millbrae Road, Mountain View, California 94031.

Story Of Javascript

- Netscape Navigator became the leading web browser.



Internet Explorer

- At the same time Microsoft Launched internet explorer to compete in the market.



Story of Javascript

- Andressen decided to push new features to Navigator to maintain the dominance in the market.
- He thought of moving Move beyond displaying static documents and become truly interactive.



Story Of Javascript

- He needs a language that could be directly written in an HTML document and will interpreted by the Navigator browser itself.

Story of Javascript

- Then, In 1995 Netscape gave this contract to Brendan Eich (deadline of 10 days).



1995 / FAST SLIM CORRECT

BRENDAN EICH DESIGNS **JAVASCRIPT** ...IN 10 DAYS

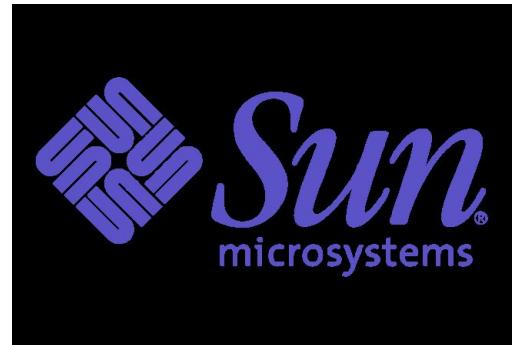


Joins NETSCAPE in April 1995.

A graphic featuring a portrait of Brendan Eich in the foreground. Behind him is a stylized globe with red, yellow, and blue swirling patterns. To the right, there is text in a bold, sans-serif font: "1995 / FAST SLIM CORRECT", "BRENDAN EICH DESIGNS **JAVASCRIPT** ...IN 10 DAYS", and "Joins NETSCAPE in April 1995". The word "JAVASCRIPT" is written in a large, yellow, blocky font.

Story of Javascript

- They want a language for the browser and he was told to make the language look like java **because Netscape was having an agreement with Sun Microsystems(who developed Java).**



The Birth of JavaScript

- ◆ December 1995 Netscape and Sun Microsystems, Inc. announced JavaScript.



NETSCAPE AND SUN ANNOUNCE JAVASCRIPT, THE OPEN, CROSS-PLATFORM OBJECT SCRIPTING LANGUAGE FOR ENTERPRISE NETWORKS AND THE INTERNET

28 INDUSTRY-LEADING COMPANIES TO ENDORSE JAVASCRIPT AS A COMPLEMENT TO JAVA FOR EASY ONLINE APPLICATION DEVELOPMENT

MOUNTAIN VIEW, Calif. (December 4, 1995) -- Netscape Communications Corporation (NASDAQ: NSCP) and Sun Microsystems, Inc. (NASDAQ:SUNW), today announced JavaScript, an open, cross-platform object scripting language for the creation and customization of applications on enterprise networks and the Internet. The JavaScript language complements Java, Sun's industry-leading object-oriented, cross-platform programming language. The initial version of JavaScript is available now as part of the beta version of Netscape Navigator 2.0, which is currently available for downloading from Netscape's web site.

In addition, 28 industry-leading companies, including America Online, Inc., Apple Computer, Inc., Architext Software, Attachmate Corporation, AT&T, Borland International, Brio Technology, Inc., Computer Associates, Inc., Digital Equipment Corporation, Hewlett-Packard Company, Ikonovex Corporation, Illustra Information Technologies, Inc., Informix Software, Inc., Intuit, Inc., Macromedia, Metrowerks, Inc., Novell, Inc., Oracle Corporation, Paper Software, Inc., Precept Software, Inc., RAD Technologies, Inc., The Santa Cruz Operation, Inc., Silicon Graphics, Inc., Spider Technologies, Sybase, Inc., Toshiba Corporation, Verity, Inc., and Vermeer Technologies, Inc., have endorsed JavaScript as an open standard object scripting language and intend to provide it in future products. The draft specification of JavaScript, as well as the final draft specification of Java, is planned for publishing and submission to appropriate standards bodies for industry review and comment this month.

JavaScript is an easy-to-use object scripting language designed for creating live online applications that link together objects and resources on both clients and servers. While Java is used by programmers to create new objects and applets, JavaScript is designed for use by HTML page authors and enterprise application developers to dynamically script the behavior of objects running on either the client or the server. JavaScript is analogous to Visual Basic in that it can be used by people with little or no programming experience to quickly construct complex applications. JavaScript's design represents the next generation of software designed specifically for the Internet and is:

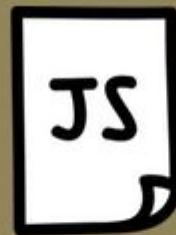
- designed for creating network-centric applications
- complementary to and integrated with Java
- complementary to and integrated with HTML
- open and cross-platform.

Story of Javascript

- Finally Eich created the language that met those requirements simple, 'looks like java'.
- Initially Its name was **Mocha** then later renamed to **LiveScript** then **Javascript**.

Story of Javascript

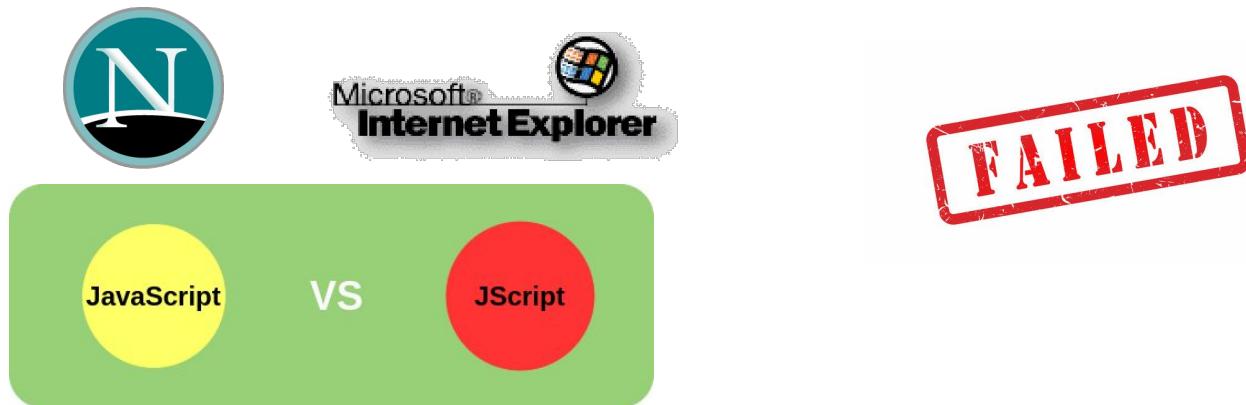
FACT



JavaScript was initially called LiveScript.
The name was changed to ride on the popularity of Java.

Microsoft - JScript

- Microsoft reverse engineer javascript support into internet explorer as Jscript in 1996 but the attempt was not perfect.



Story of Javascript

- In 1997, Netscape realized it should be consistent across browser.
- So they decided to standardize the language under an International standards organization called ECMA meaning European Computer Manufacturers' Association.

JavaScript was officially referred to as ECMAScript.



Story of Javascript

- ECMAScript underwent a lot of revisions and improvements. ES2(1998) and ES3(1999) included a good number of features.
- But still, Javascript faced a lot of setbacks from different companies developers.



Ajax

- In 2005 , Jesse James bring AJAX and Javascript popularize.
- Ajax improve the user experience.



Jquery

- Later Introducing Jquery a powerful library of JS which again revive the popularity of javascript.
- Many new versions of EcmaScript keep on releasing till today.

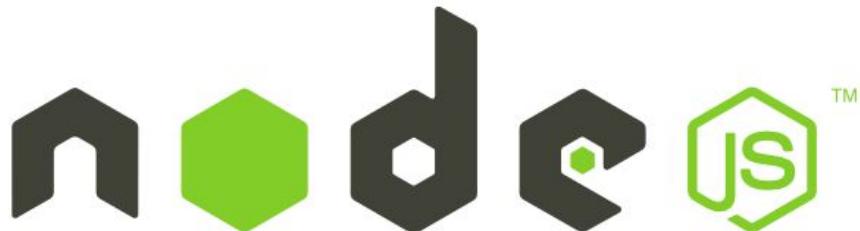


Ecma Releases



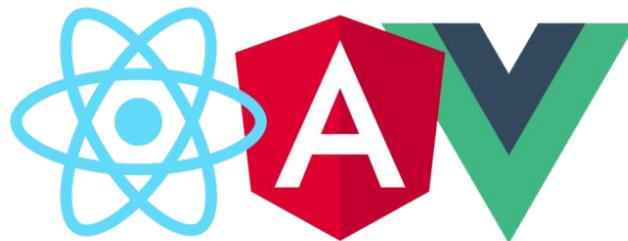
Node.js

- Till today we only use the javascript on the client-side but the later introduction of Node.js, through which we able to run javascript on the servers.



Javascript Frameworks

- A series of JavaScript frameworks and libraries, such as Ember, Angular, React, and Vue, have been developed to allow powerful and complicated web applications to be written using small teams within short time spans.





Most Popular Technologies

Programming, Scripting, and Markup Languages

All Respondents

Professional Developers

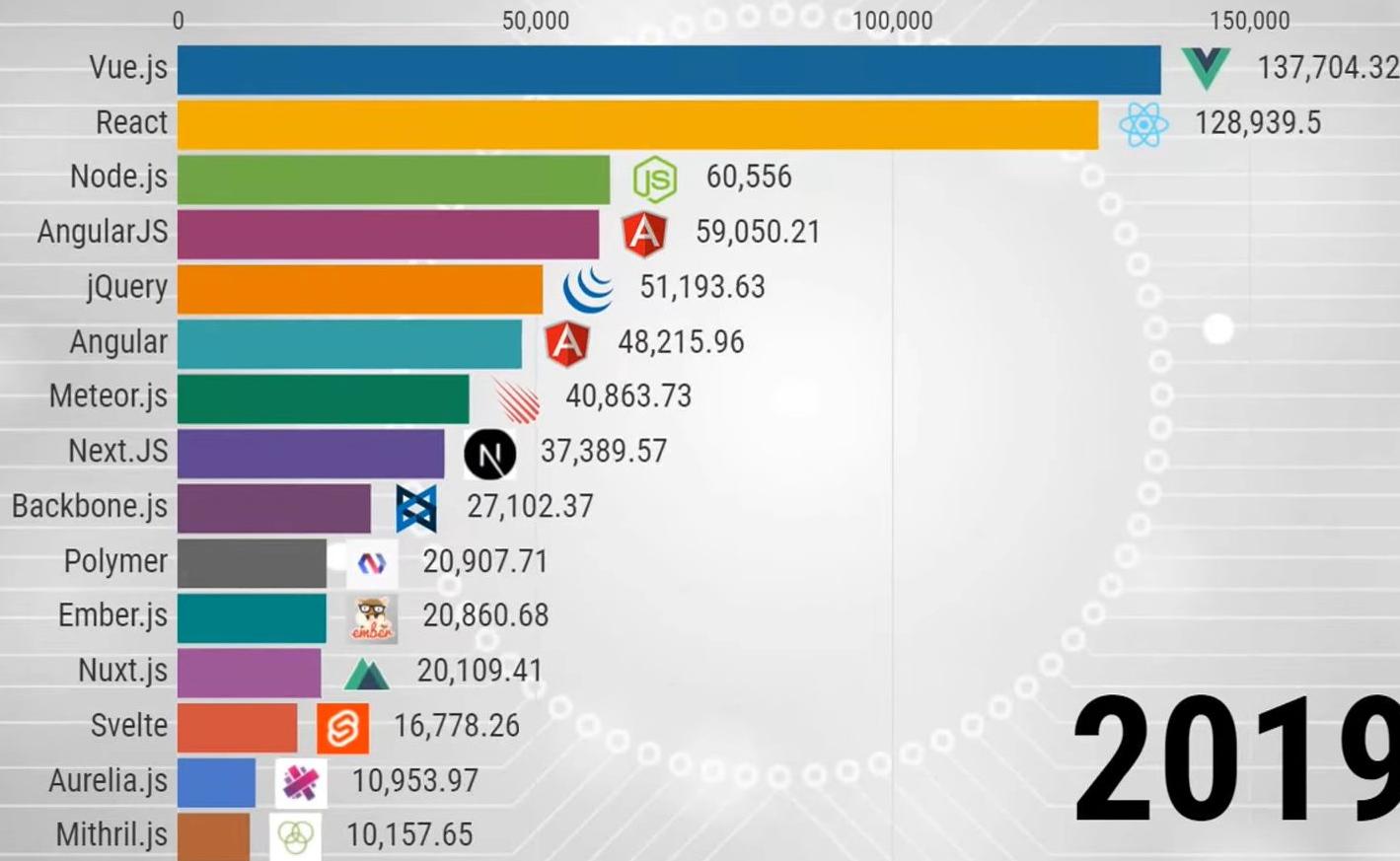


Popularity

- JavaScript has risen to be the most popular programming language in the world. According to GitHub's 2018 Octoverse report,



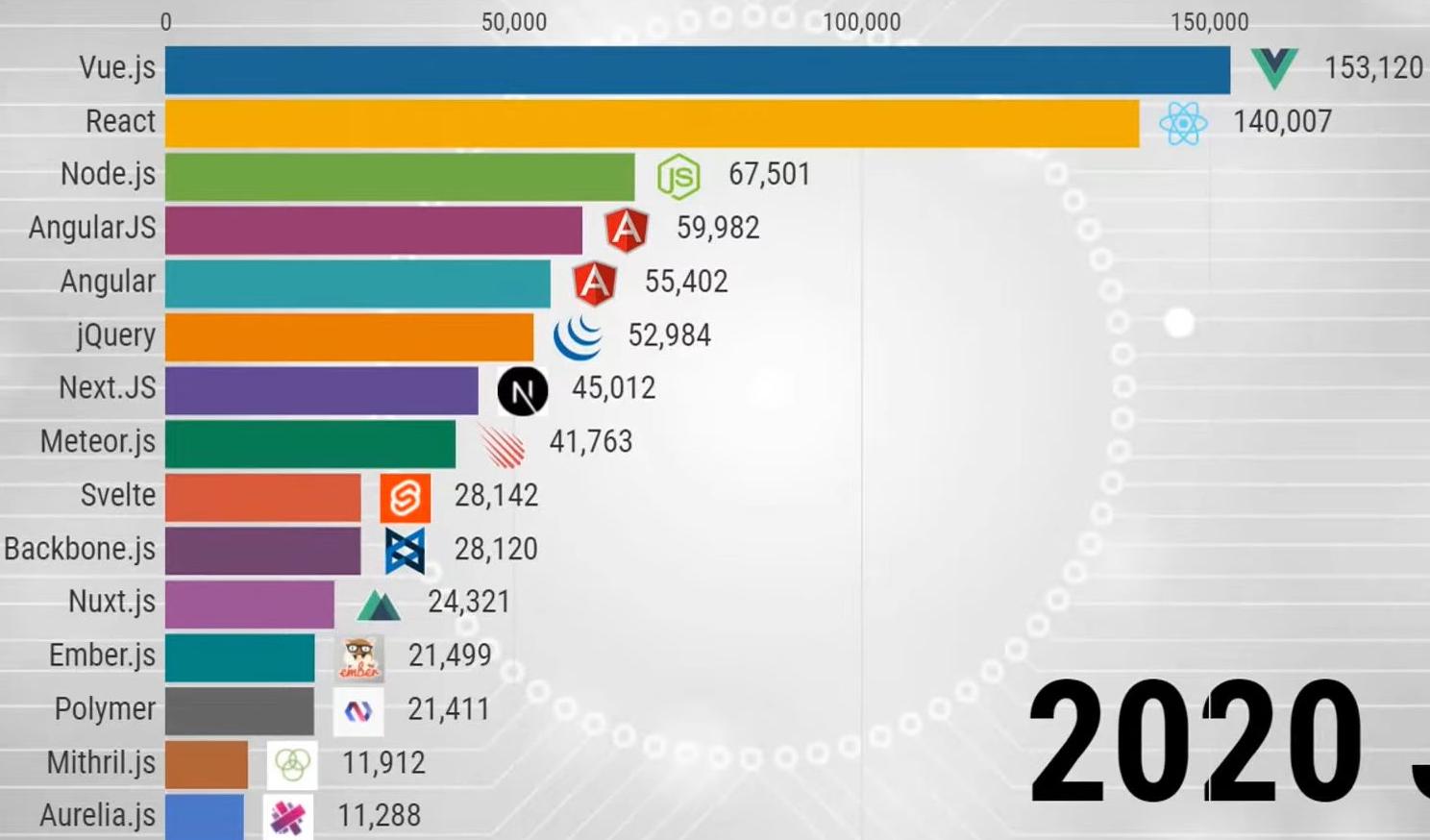
Most Popular JavaScript Frameworks



2019 Apr



Most Popular JavaScript Frameworks



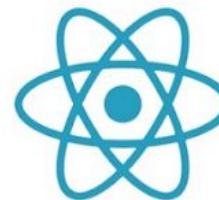
2020 Jan



Meteor JS



Ember JS



React JS

masai®



Vue JS



Polymer JS



Best JavaScript Frameworks



Angular JS

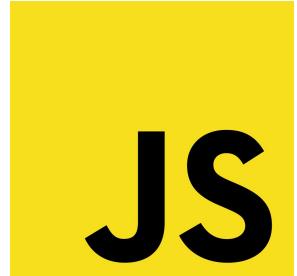
GG

Always bet on
JavaScript

- **Brendan Eich**

Javascript

- Javascript is a client side language means it runs on your computer within your browser.
- Node.js allowed javascript to also execute code on servers.

 JS

Structure of a website



HTML
Structure



CSS
Presentation



JAVASCRIPT
Behavioral

THE ROLE OF HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:100,300,400,600" rel="stylesheet"
    type="text/css">
    <link href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" rel="stylesheet"
    type="text/css">
    <link type="text/css" rel="stylesheet" href="style.css">
    <title>Budgety</title>
  </head>
  <body>

    <div class="top">
      <div class="budget">
        <div class="budget__title">
          Available Budget in <span class="budget__title--month">3 Month</span>
        </div>

        <div class="budget__value">+ 2,345.64</div>

        <div class="budget__income clearfix">
          <div class="budget__income--text">Income</div>
          <div class="right">
            <div class="budget__income--value">+ 4,300.00</div>
            <div class="budget__income--percentage">&ampnbsp</div>
          </div>
        </div>

        <div class="budget__expenses clearfix">
          <div class="budget__expenses--text">Expenses</div>
          <div class="right clearfix">
            <div class="budget__expenses--value">- 1,954.36</div>
            <div class="budget__expenses--percentage">45%</div>
          </div>
        </div>
      </div>
    </div>

    <div class="bottom">
      <div class="add">
        <div class="add__container">
          <select class="add__type">
            <option value="inc" selected>+</option>
            <option value="exp">-</option>
          </select>
          <input type="text" class="add__description" placeholder="Add description">
          <input type="number" class="add__value" placeholder="Value">
          <button class="add__btn"><i class="ion-ios-checkmark-outline"></i></button>
        </div>
      </div>
    </div>
```

THE ROLE OF HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:100,300,400,600" rel="stylesheet"
    type="text/css">
    <link href="http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" rel="stylesheet"
    type="text/css">
    <link type="text/css" rel="stylesheet" href="style.css">
    <title>Budgety</title>
  </head>
  <body>

    <div class="top">
      <div class="budget">
        <div class="budget__title">
          Available Budget in <span class="budget__title--month">%Month%</span>:
        </div>

        <div class="budget__value">+ 2,345.64</div>

        <div class="budget__income clearfix">
          <div class="budget__income--text">Income</div>
          <div class="right">
            <div class="budget__income--value">+ 4,300.00</div>
            <div class="budget__income--percentage">$nbsp;</div>
          </div>
        </div>

        <div class="budget__expenses clearfix">
          <div class="budget__expenses--text">Expenses</div>
          <div class="right clearfix">
            <div class="budget__expenses--value">- 1,954.36</div>
            <div class="budget__expenses--percentage">45%</div>
          </div>
        </div>
      </div>
    </div>

    <div class="bottom">
      <div class="add">
        <div class="add__container">
          <select class="add__type">
            <option value="inc" selected>+</option>
            <option value="exp">-</option>
          </select>
          <input type="text" class="add__description" placeholder="Add description">
          <input type="number" class="add__value" placeholder="Value">
          <button class="add__btn"><i class="ion-ios-checkmark-outline"></i></button>
        </div>
      </div>
    </div>
  </body>
</html>
```



Available Budget in January 2018:

+ 175.00
Income
+ 3,000.00

Expenses

- 2,825.00
94%

+ ⚡ Add description Value ✎

Income

Salary
+ 2,500.00
⊗
Rent
+ 500.00
⊗

Expenses

Taxes
- 1,500.00
50%
⊗
Groceries
- 450.00
15%
⊗
Holidays
- 875.00
29%
⊗

```
.budget {  
    position: absolute;  
    width: 350px;  
    top: 50%;  
    left: 50%;  
    transform: translate(-50%, -50%);  
    color: #fff;  
}  
  
.budget__title {  
    font-size: 18px;  
    text-align: center;  
    margin-bottom: 10px;  
    font-weight: 300;  
}  
  
.budget__value {  
    font-weight: 300;  
    font-size: 46px;  
    text-align: center;  
    margin-bottom: 25px;  
    letter-spacing: 2px;  
}  
  
.budget__income,  
.budget__expenses {  
    padding: 12px;  
    text-transform: uppercase;  
}  
  
.budget__income {  
    margin-bottom: 10px;  
    background-color: #28B985;  
}  
  
.budget__expenses {  
    background-color: #FF5049;  
}  
  
.budget__income--text,  
.budget__expenses--text {  
    float: left;  
    font-size: 13px;  
    color: #444;  
    margin-top: 2px;  
}
```

THE ROLE OF CSS

```
.budget {  
  position: absolute;  
  width: 350px;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
  color: #fff;  
}  
  
.budget__title {  
  font-size: 18px;  
  text-align: center;  
  margin-bottom: 10px;  
  font-weight: 300;  
}  
  
.budget__value {  
  font-weight: 300;  
  font-size: 46px;  
  text-align: center;  
  margin-bottom: 25px;  
  letter-spacing: 2px;  
}  
  
.budget__income,  
.budget__expenses {  
  padding: 12px;  
  text-transform: uppercase;  
}  
  
.budget__income {  
  margin-bottom: 10px;  
  background-color: #28B985;  
}  
  
.budget__expenses {  
  background-color: #FF5049;  
}  
  
.budget__income--text,  
.budget__expenses--text {  
  float: left;  
  font-size: 13px;  
  color: #444;  
  margin-top: 2px;  
}
```

Available Budget in January 2018:
+ 175.00
Income
+ 3,000.00

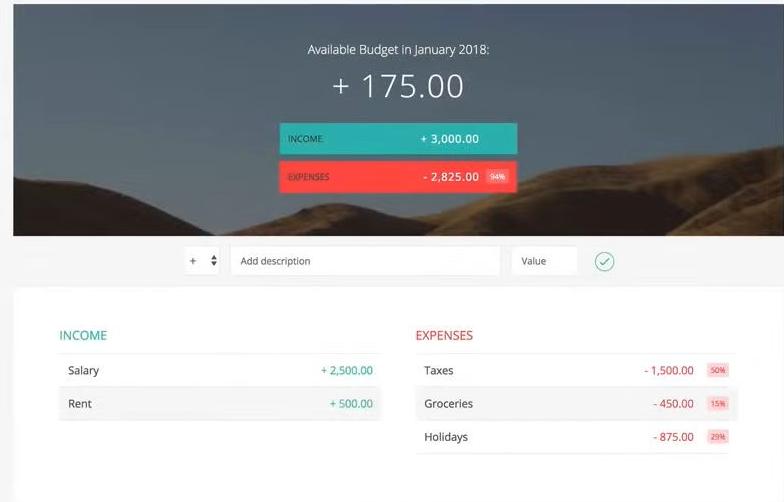
Expenses
- 2,825.00
94%
• Add description Value

Income

Salary	+ 2,500.00
Rent	+ 500.00

Expenses

Taxes	- 1,500.00
Groceries	- 450.00
Holidays	- 875.00



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

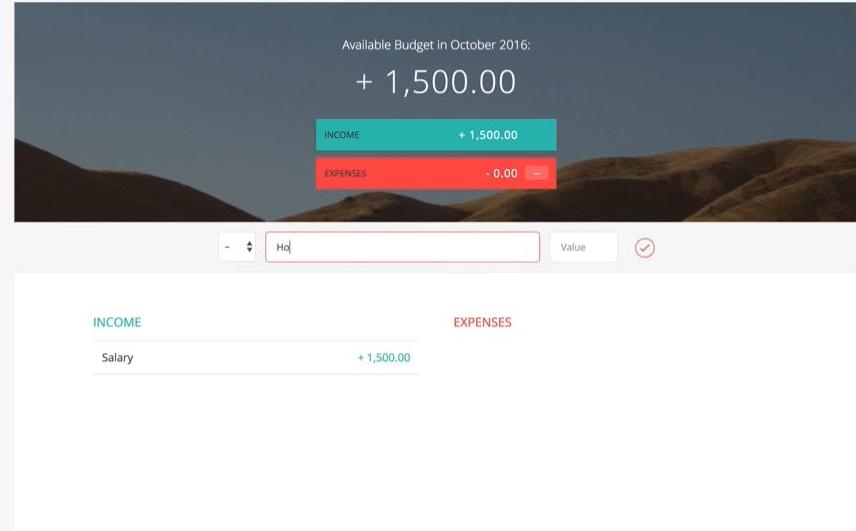
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

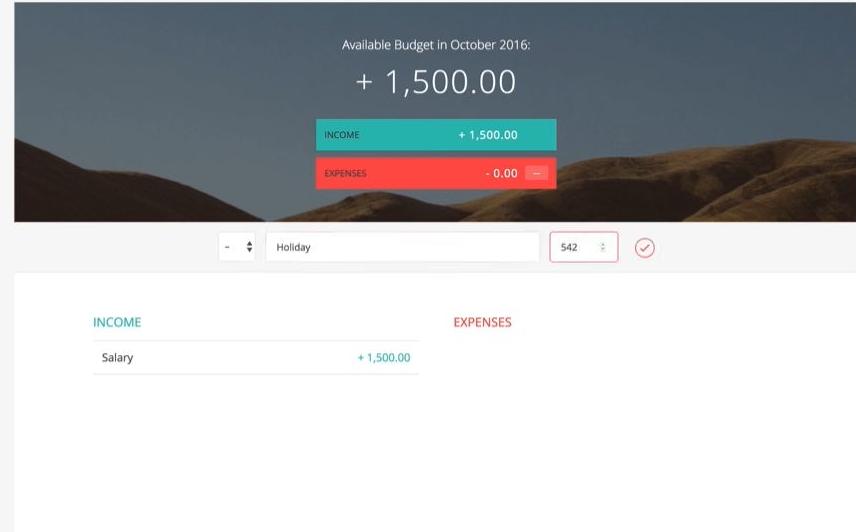
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

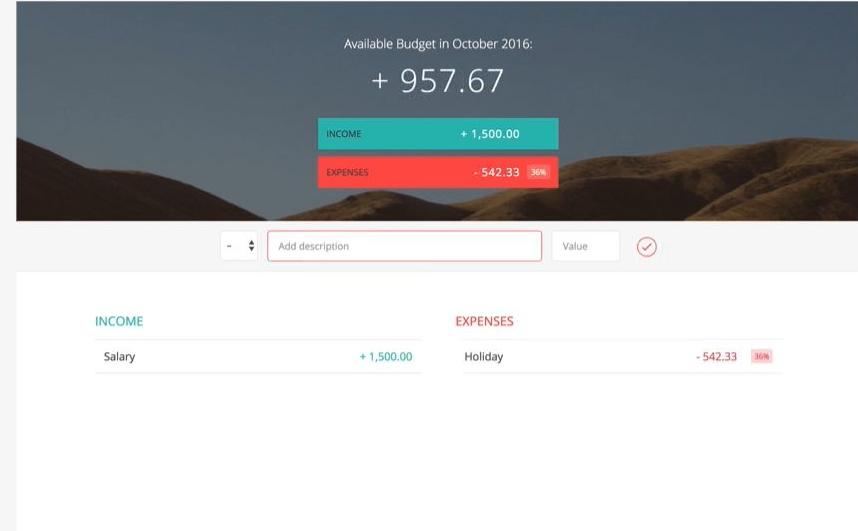
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

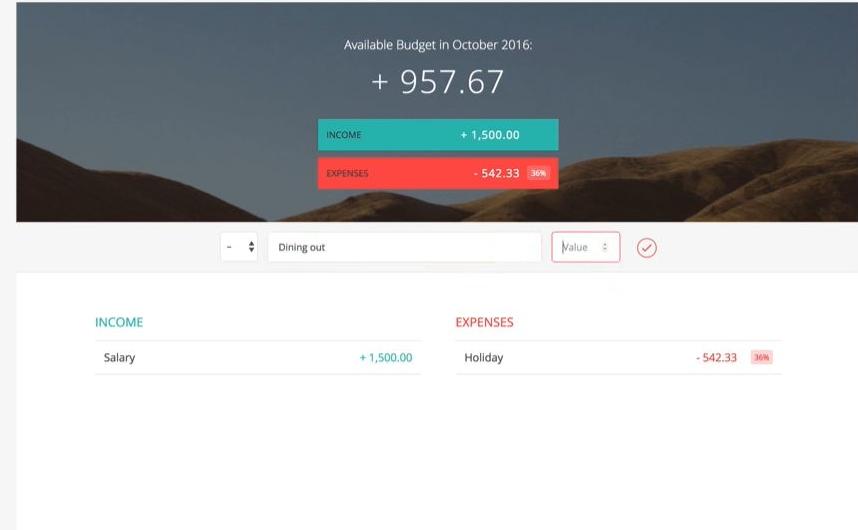
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

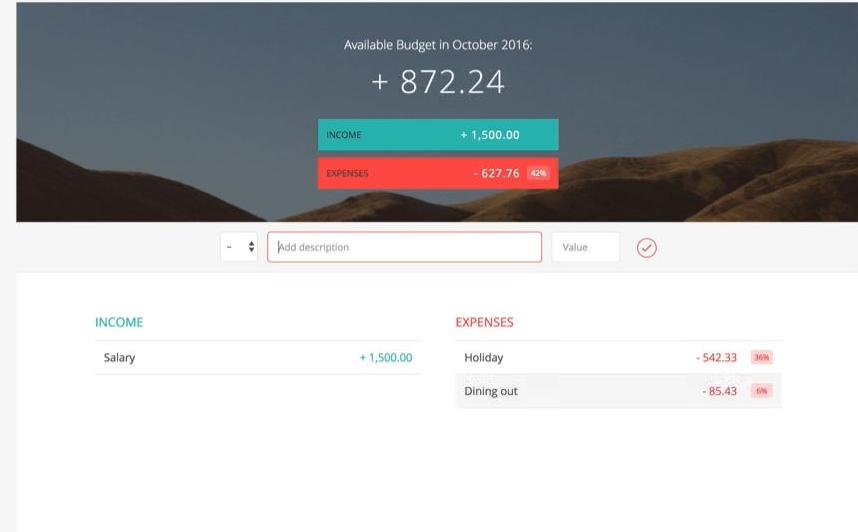
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

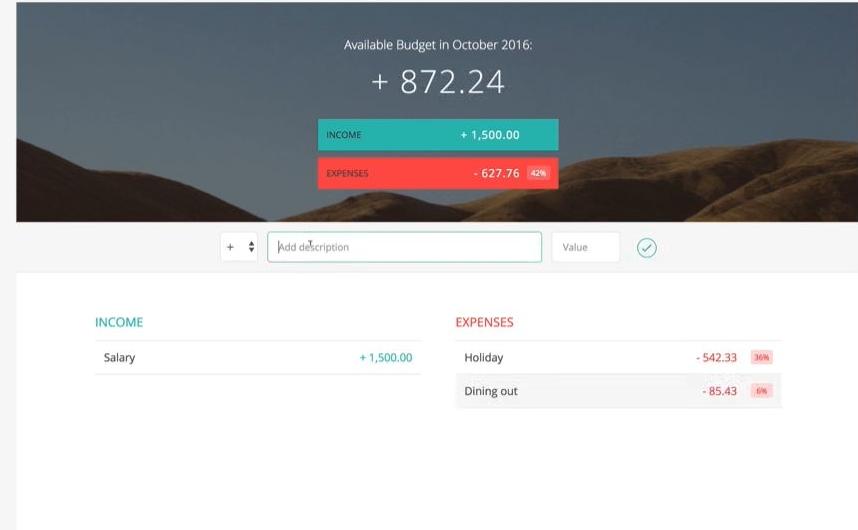
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

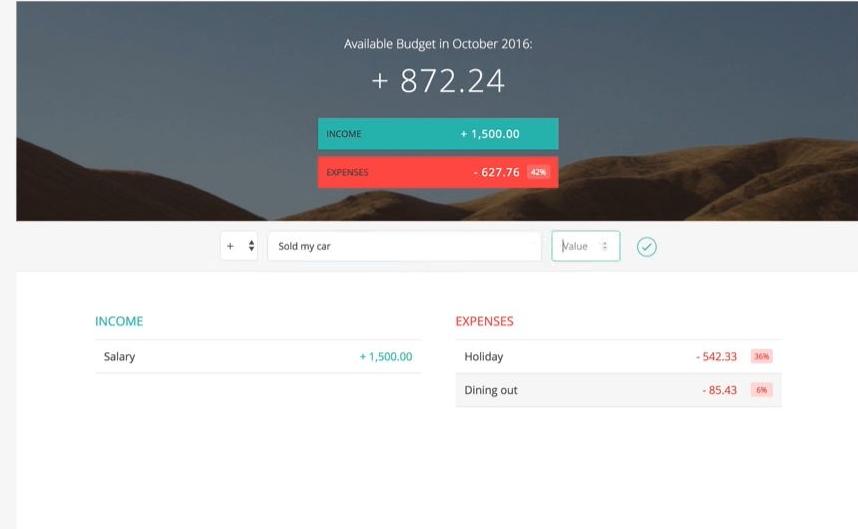
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

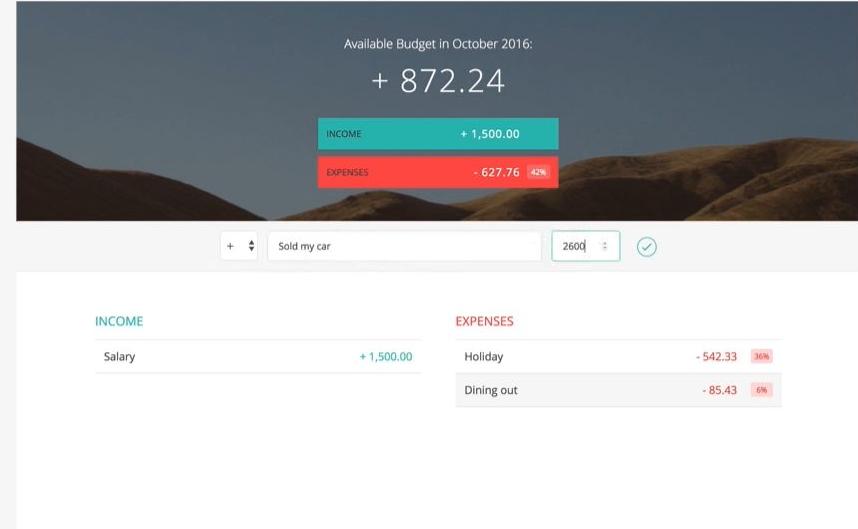
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

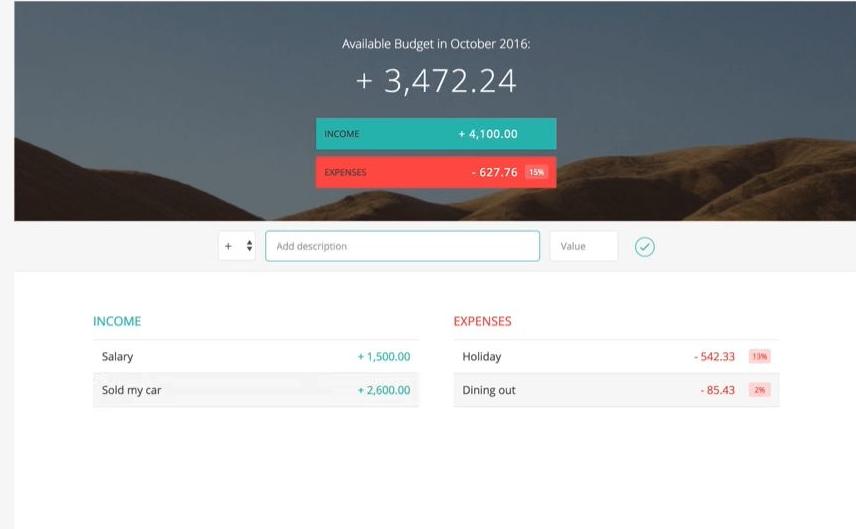
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

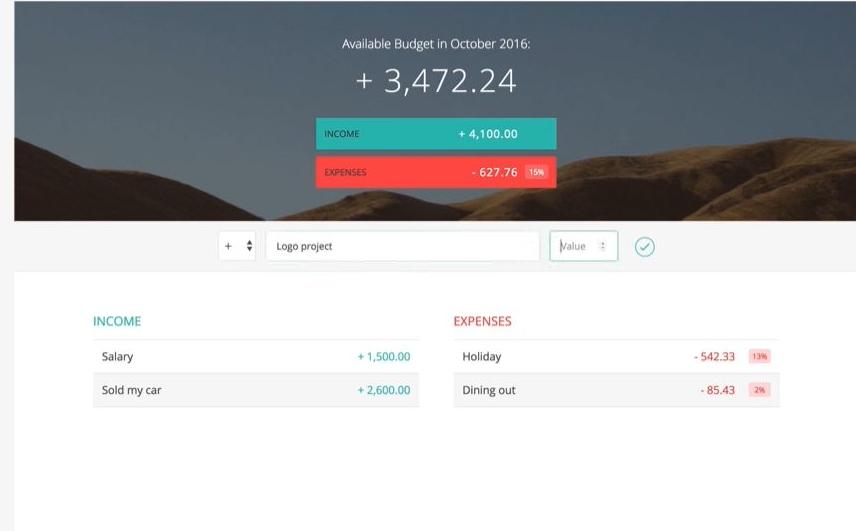
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

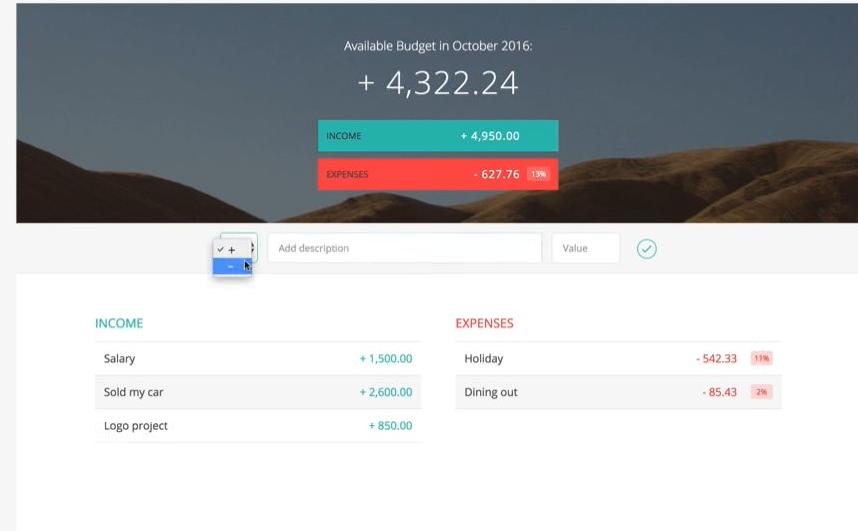
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

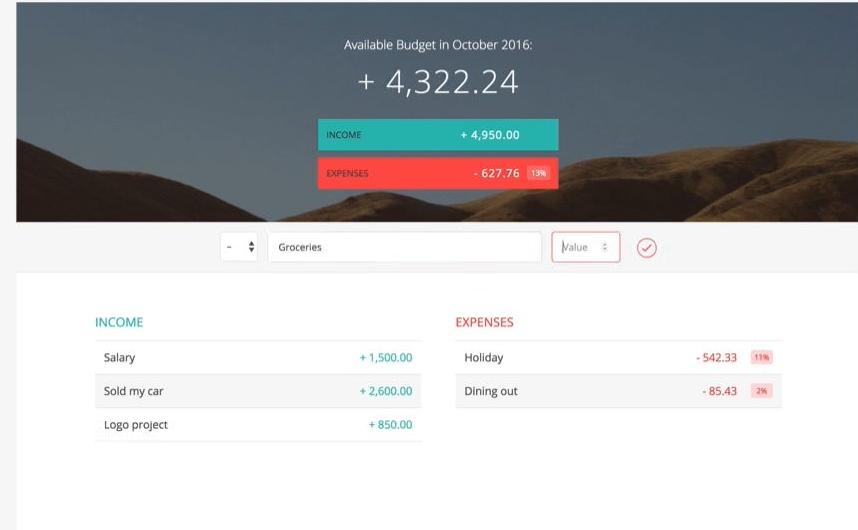
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UIctrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

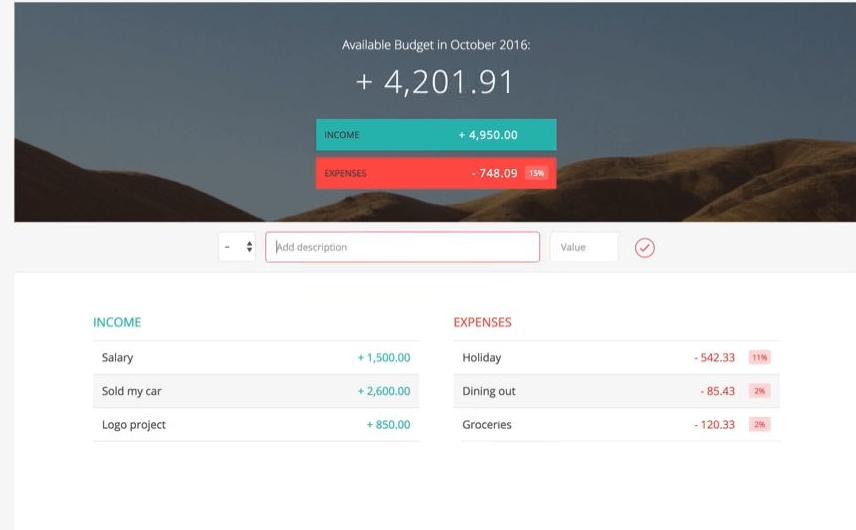
var setupEventListeners = function() {
  var DOM = UIctrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UIctrl.changedType);
};
```



THE ROLE OF JAVASCRIPT

```
var ctrlDeleteItem = function(event) {
  var clickID, splitID, type, id;
  clickID = event.target.parentNode.parentNode.parentNode.id;

  if (clickID) {
    splitID = clickID.split('-');
    type = splitID[0];
    id = parseInt(splitID[1]);

    // 1. Delete item from model
    if (type === 'expense') {
      budgetCtrl.deleteItem('exp', id);
    } else {
      budgetCtrl.deleteItem('inc', id);
    }

    // 2. Delete item from UI
    UICtrl.deleteListItem(clickID);

    // 3. Update and show the new totals
    updateBudget();

    // 4. Update and show all expense percentages
    updatePercentages();
  }
}

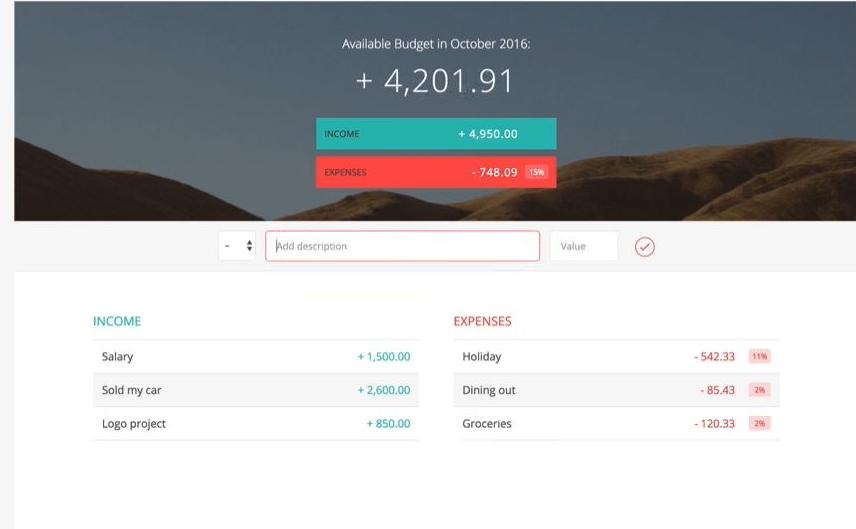
var setupEventListeners = function() {
  var DOM = UICtrl.getDOMstrings();

  // Click on input button
  document.querySelector(DOM.inputBtn).addEventListener('click', ctrlAddItem);

  // Hit enter on input form
  document.addEventListener('keypress', function(event) {
    if (event.which === 13 || event.keyCode === 13) {
      ctrlAddItem();
    }
  });

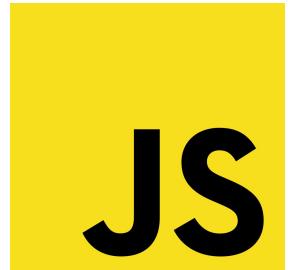
  // Click delete icon
  document.querySelector(DOM.container).addEventListener('click', ctrlDeleteItem);

  // Change expense/income [LATER]
  document.querySelector(DOM.inputType).addEventListener('change', UICtrl.changedType);
};
```



Javascript

- HTML and CSS give a website structure and style.
- Javascript add functionality and behaviours to the website, which allow website visitor to interact with content.

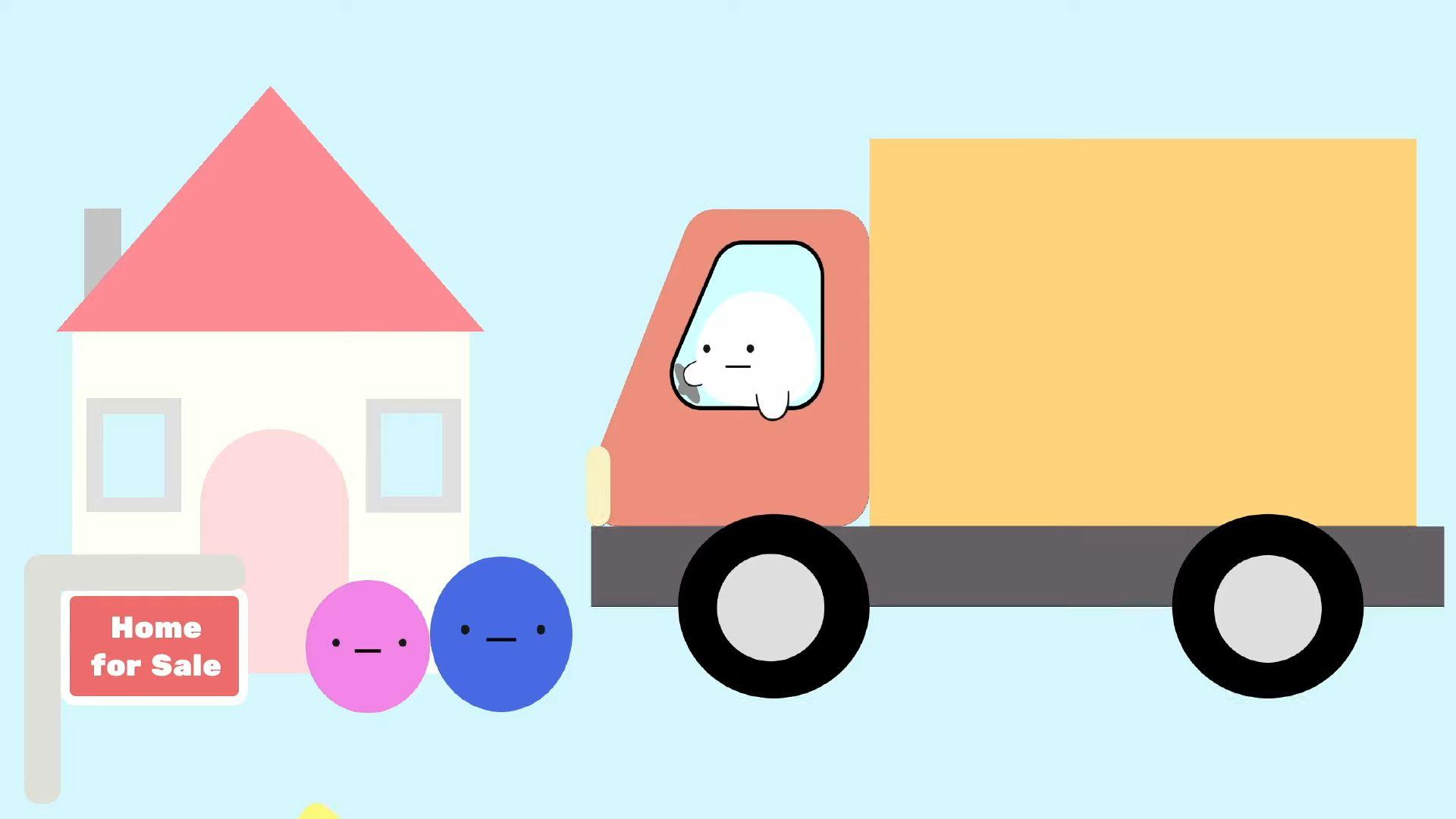




Javascript

- Visit website <https://codepen.io/rctyou/pen/QEObEk/>

Variables



Home
for Sale





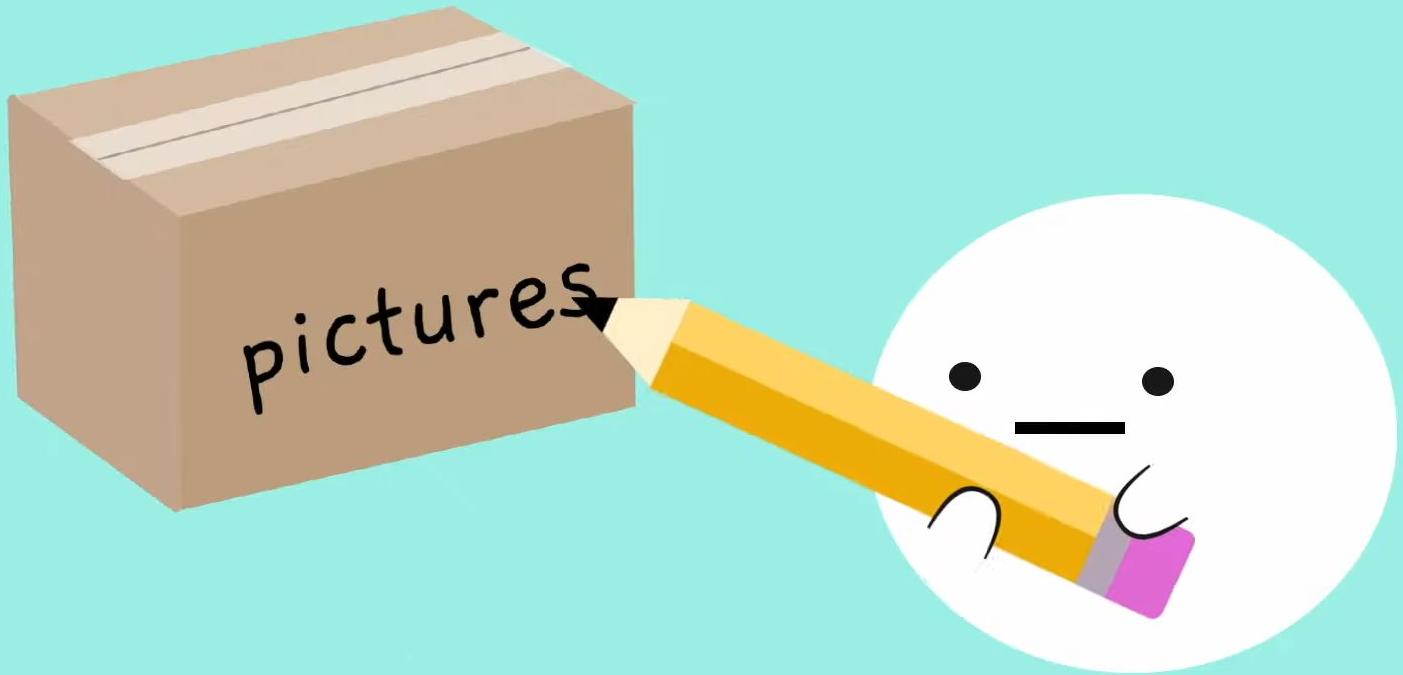


=

variable

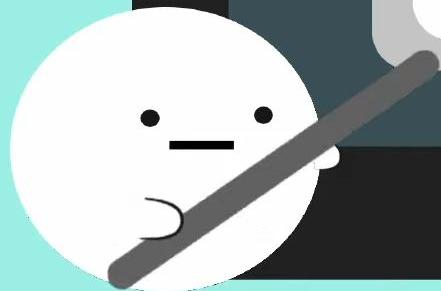






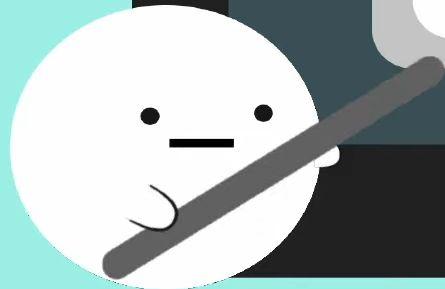
Variables

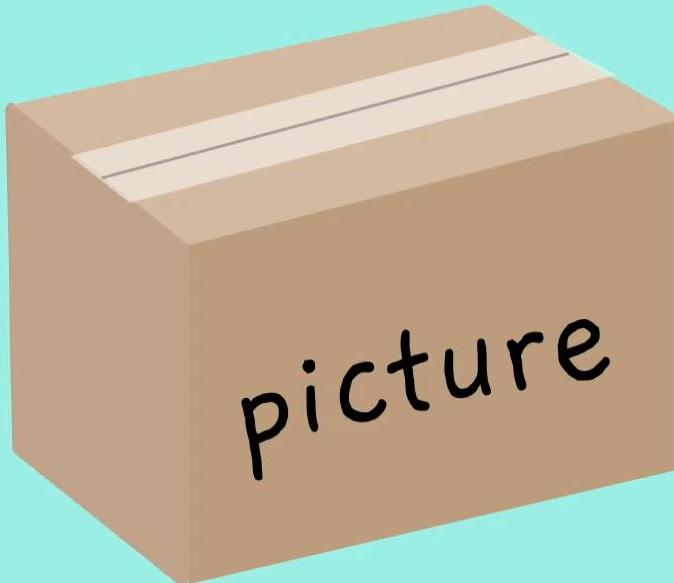
Storage locations
with assigned names



Variables

Hold data that you can
change or use later







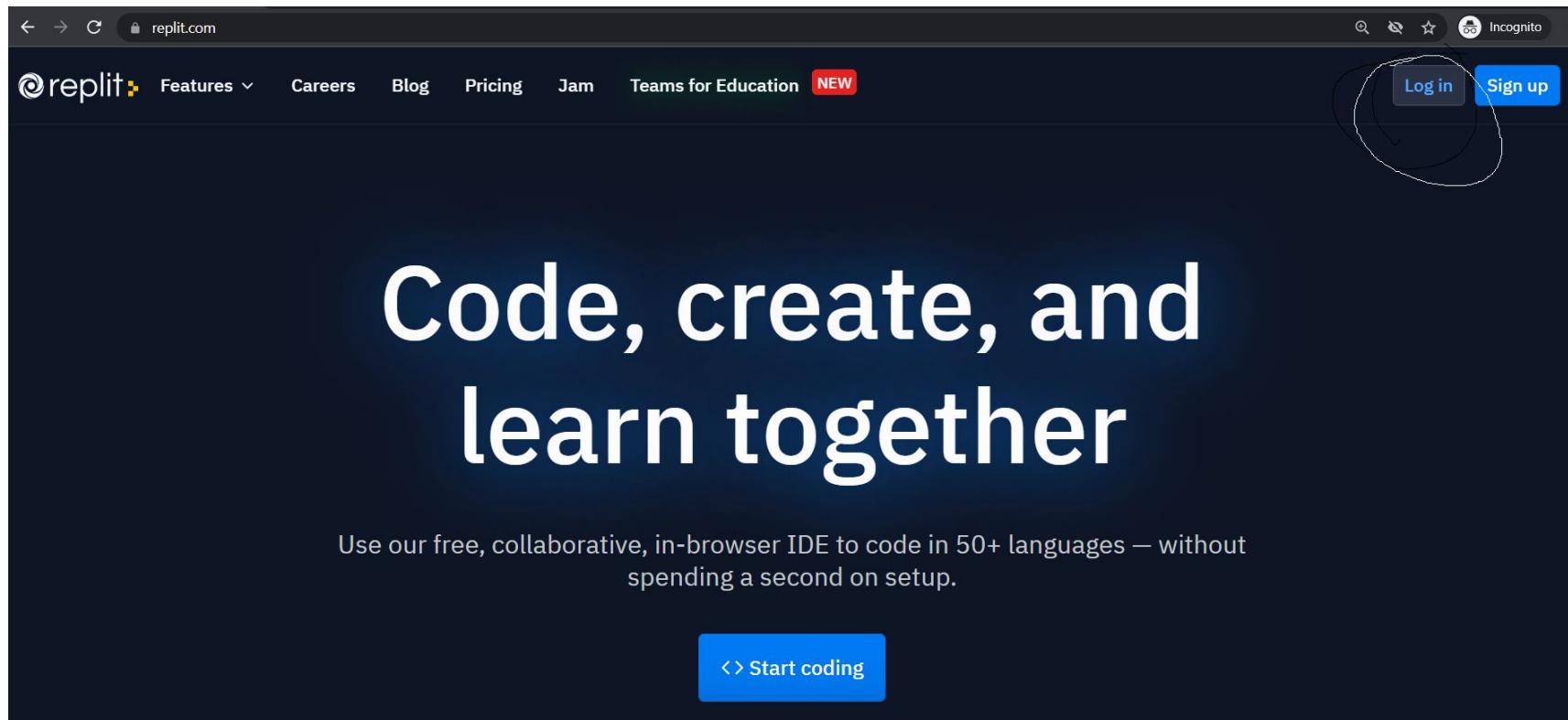
Setup your Coding Environment



Setup Replit



Step 1 : Go to replit.com and login



A screenshot of a web browser displaying the Replit homepage. The URL 'replit.com' is visible in the address bar. The page features a dark blue header with navigation links: '@replit' (with a yellow gear icon), 'Features', 'Careers', 'Blog', 'Pricing', 'Jam', 'Teams for Education' (with a 'NEW' badge), and 'Log in' (highlighted with a hand-drawn circle) and 'Sign up'. Below the header, a large white text area reads 'Code, create, and learn together'. A subtext below states: 'Use our free, collaborative, in-browser IDE to code in 50+ languages — without spending a second on setup.' At the bottom center is a blue button with the text '<> Start coding'.

Step 2 : Press + button on top right

The screenshot shows the replit.com homepage. On the left is a sidebar with navigation links: Home (selected), Apps (BETA), My Repls, Talk, Learn, Teams (selected), Curriculum, Blog, About, Careers, Pricing, and Discord. At the top right is a search bar labeled "Search & run commands" with a "Ctrl" key icon. A large promotional overlay titled "Teams Pro" is centered. It features the text "Use replit with your team at work." and "Teams pro includes:" followed by a list of features with icons: Technical Interviewing (two people), Shared private repls (key), Extra speed & storage (cube), Comments & threads (checkmark), and Free hosting & always on repls (infinity symbol). To the right of the text is a colorful illustration of a multi-armed robot or octopus holding various items like a laptop, books, and donuts. At the bottom of the overlay is a blue button labeled "Join the waitlist" and a link "learn more". In the bottom right corner of the overlay is a black starburst badge with the text "Available Now". A hand-drawn style circle with a plus sign is drawn over the top right corner of the main workspace area.

replit.com/~

@VarunBhatt3

+ Create Repl

Upgrade

Home

Apps BETA

My Repls

Talk

Learn

Teams

Curriculum

Blog

About

Careers

Pricing

Discord

Search & run commands

Ctrl

+

Teams Pro

Use replit with your team at work.

Teams pro includes:

- Technical Interviewing
- Shared private repls
- Extra speed & storage
- Comments & threads
- Free hosting & always on repls

Join the waitlist

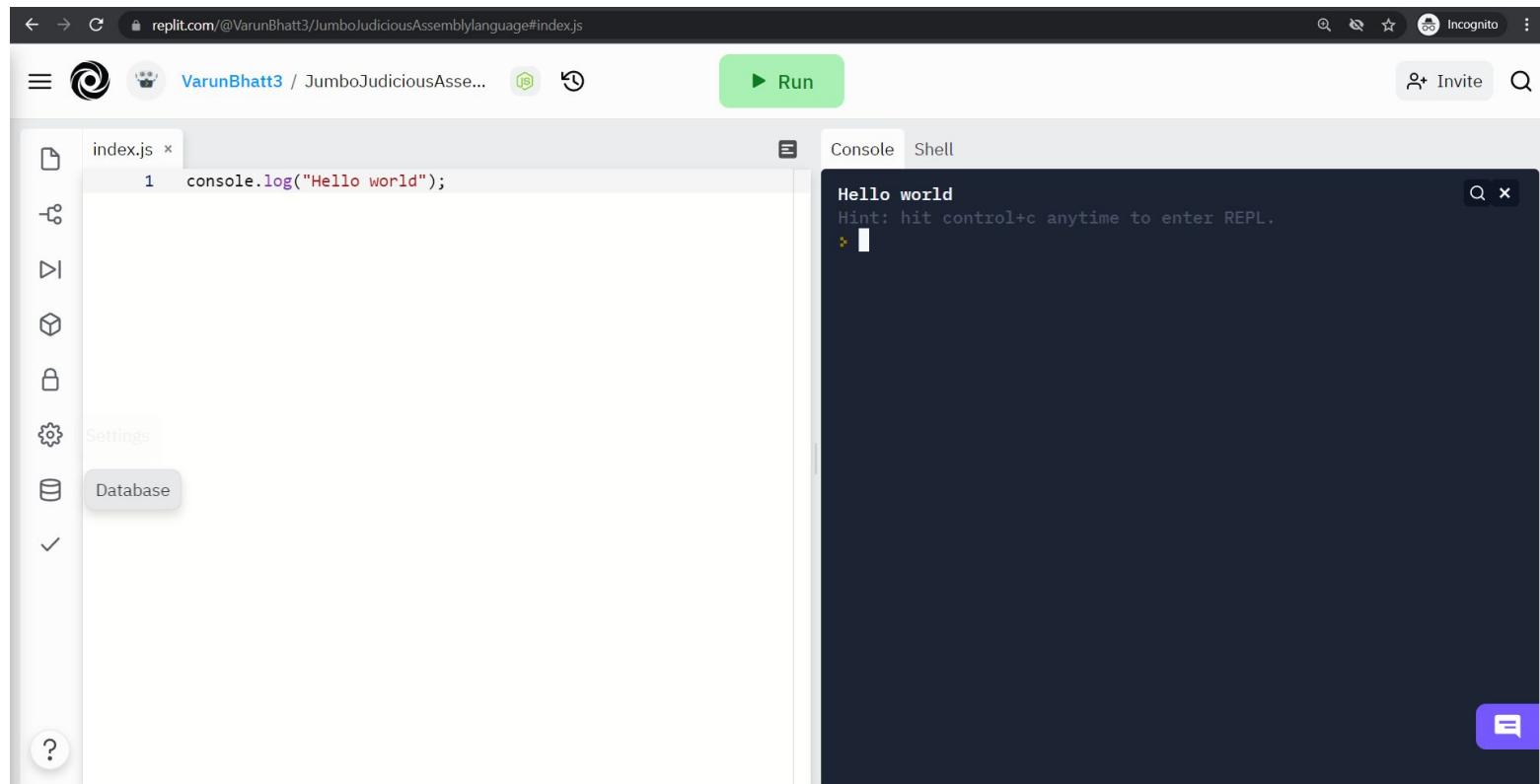
learn more

Available Now

Step 3 : Select Template Node.Js and Create Repl

The screenshot shows the Replit web interface. On the left, there's a sidebar with navigation links like Home, Apps (BETA), My Repls, Talk, Learn, Teams, Curriculum, Blog, About, Careers, Pricing, and Discord. The main area has a search bar at the top. A central modal window titled "Create a repl" is open. Inside the modal, a dropdown menu labeled "Template" is set to "Node.js". Below the template selection, there's a preview box for "Node.js" which is described as "Runtime for writing back-end JavaScript." To the right of the preview, there are sections for "Languages" and "Title" (with the value "JumboJudiciousAssemblylanguage" highlighted). There are also "Privacy" settings ("Repl is public") and a "Anyone can view and fork this repl" toggle. At the bottom of the modal is a large blue button labeled "+ Create Repl". A hand-drawn circle highlights the "Node.js" template selection. Another hand-drawn circle highlights the "+ Create Repl" button. In the bottom right corner of the screen, there's a circular badge with the text "Available Now".

Step 4 : Write code and press Run



The screenshot shows the Replit IDE interface. On the left, there's a sidebar with icons for file operations like new file, copy, paste, and delete, as well as settings, database, and help. The main workspace has a file named "index.js" open, containing the single line of code:

```
1 console.log("Hello world");
```

. To the right of the code editor is a "Run" button. Below the code editor is a dark-themed terminal window titled "Console". It displays the output of the code execution:

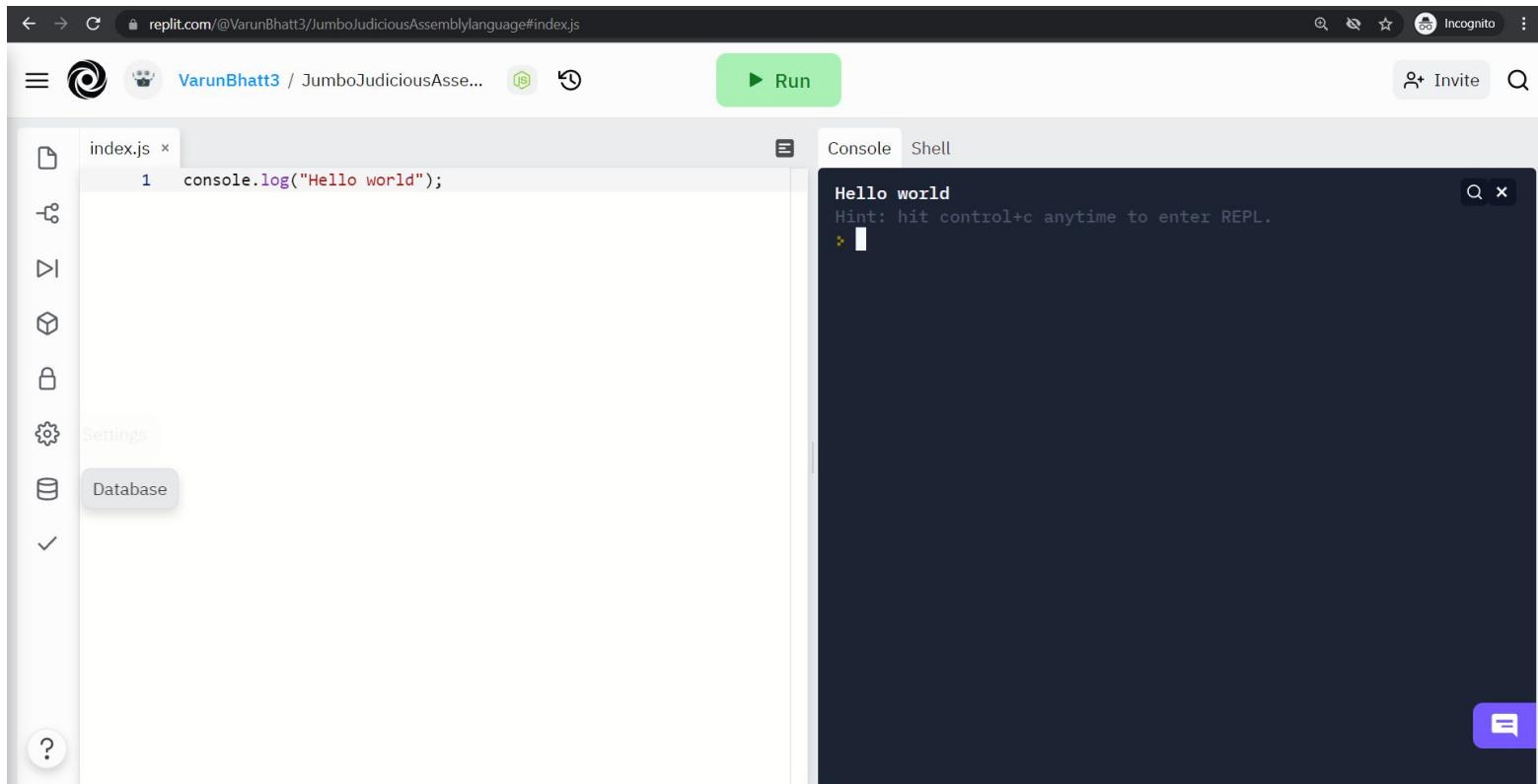
```
Hello world
```

 followed by a hint: "Hint: hit control+c anytime to enter REPL.". A small purple message icon is visible in the bottom right corner of the terminal area.

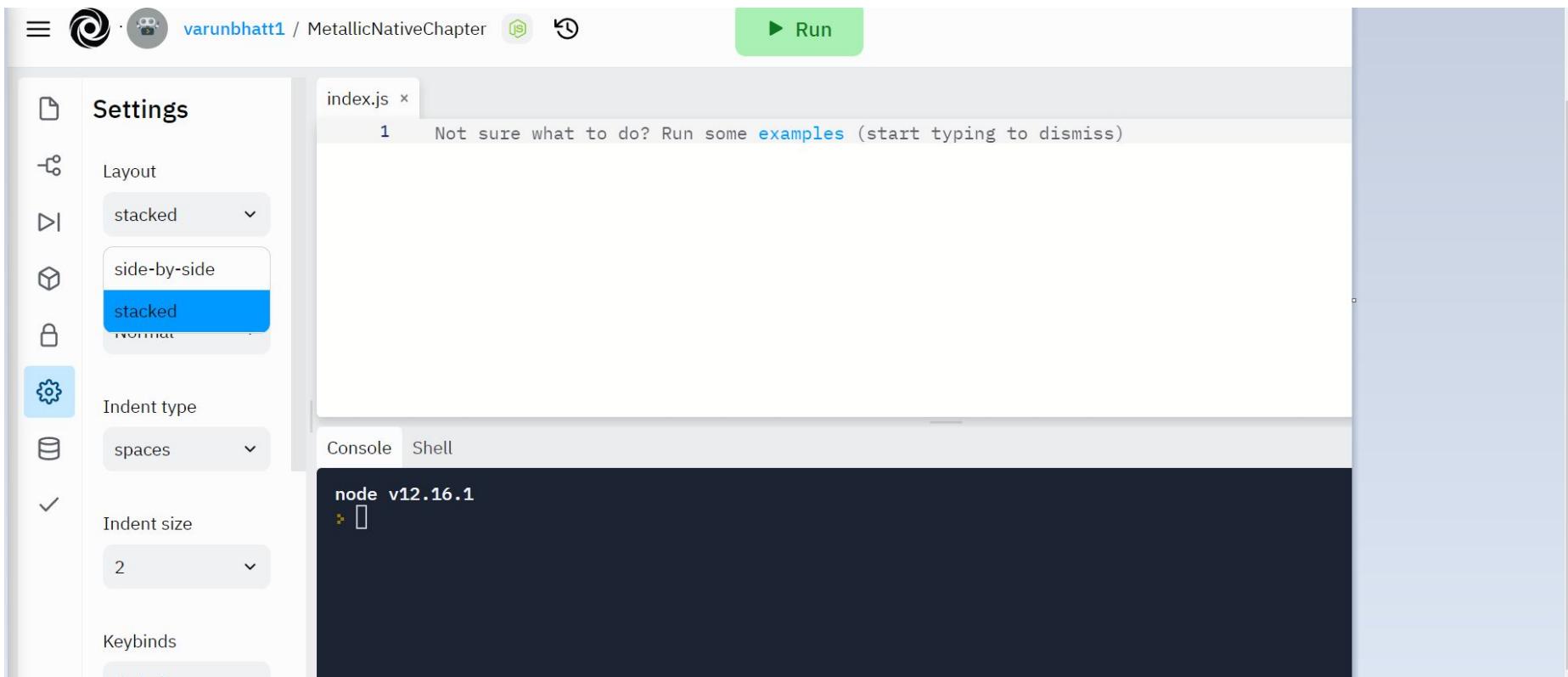


Setup “Split Screen” view

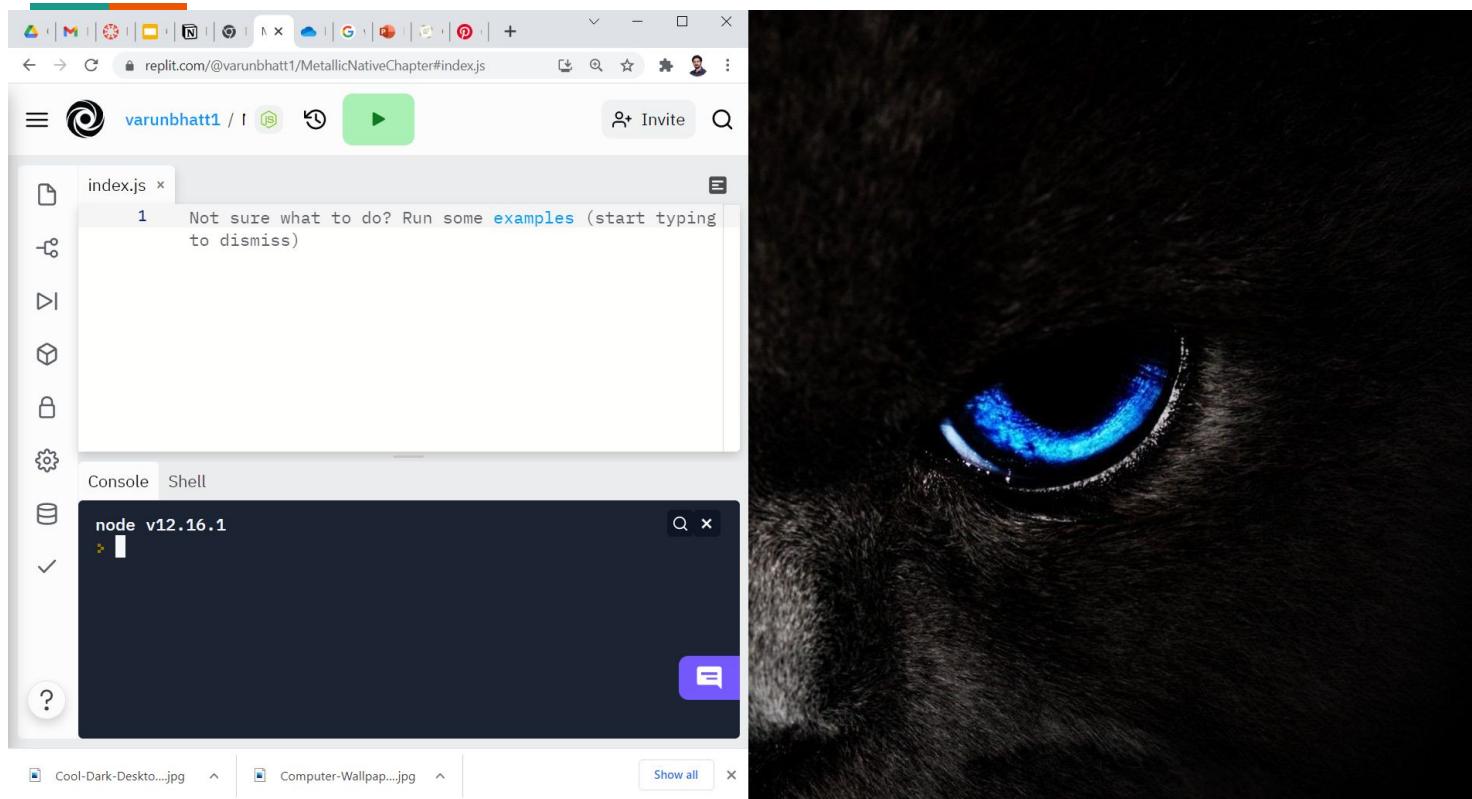
Step 1 : Open Replit



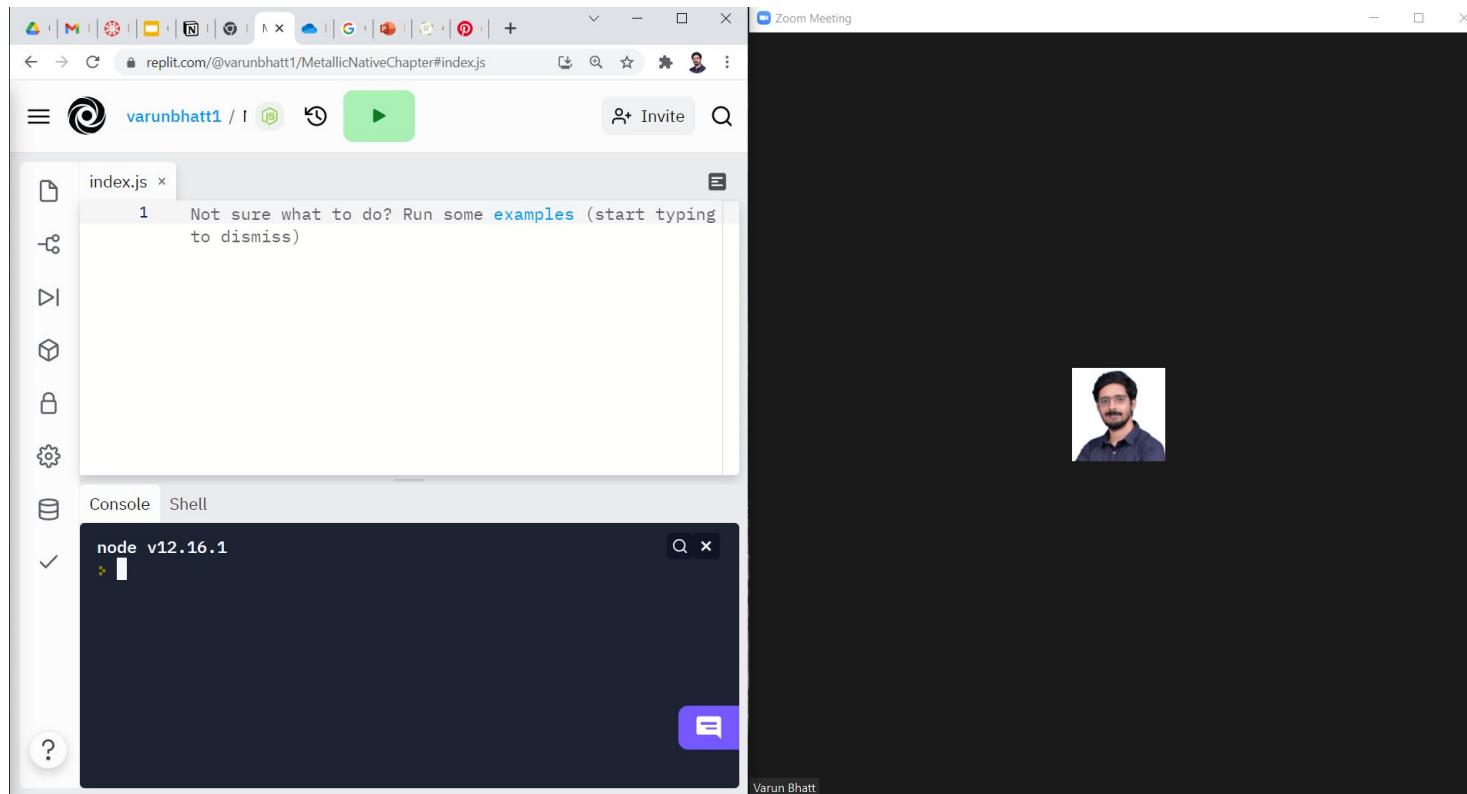
Step 2 : Choose stacked mode



Step 3 : Put the replit window in left half

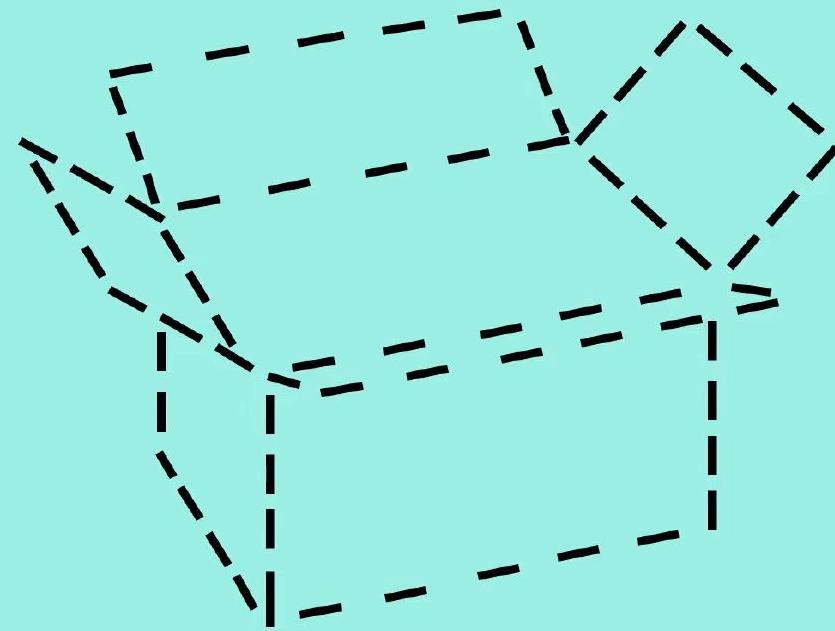
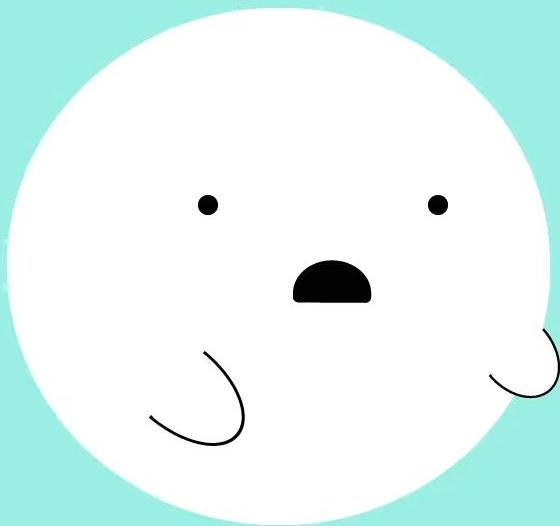


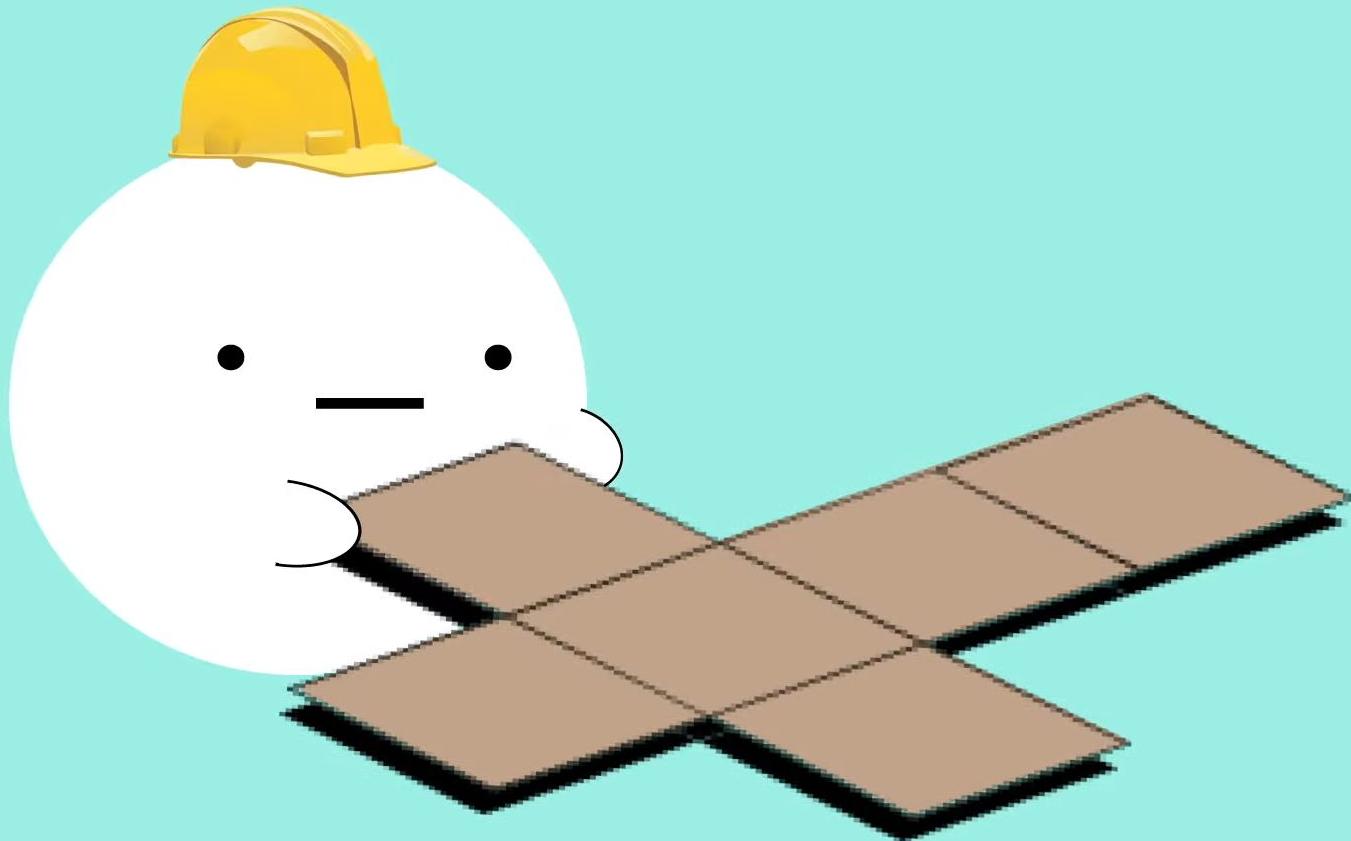
Step 4 : Put the zoom window in right half



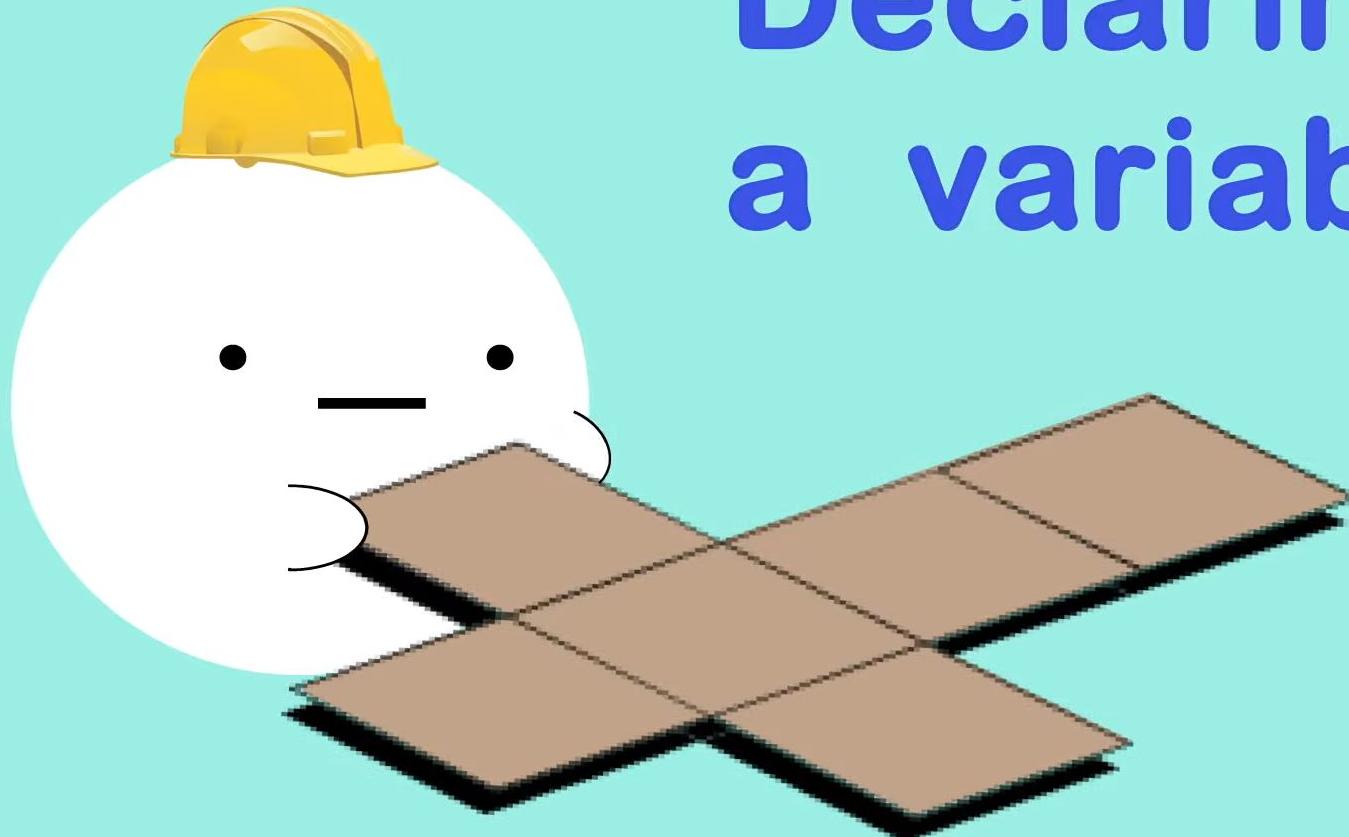


Post your split screen view setup in Discussion



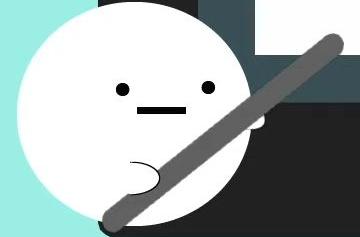


Declaring a variable





Data Type Variable Name = Value

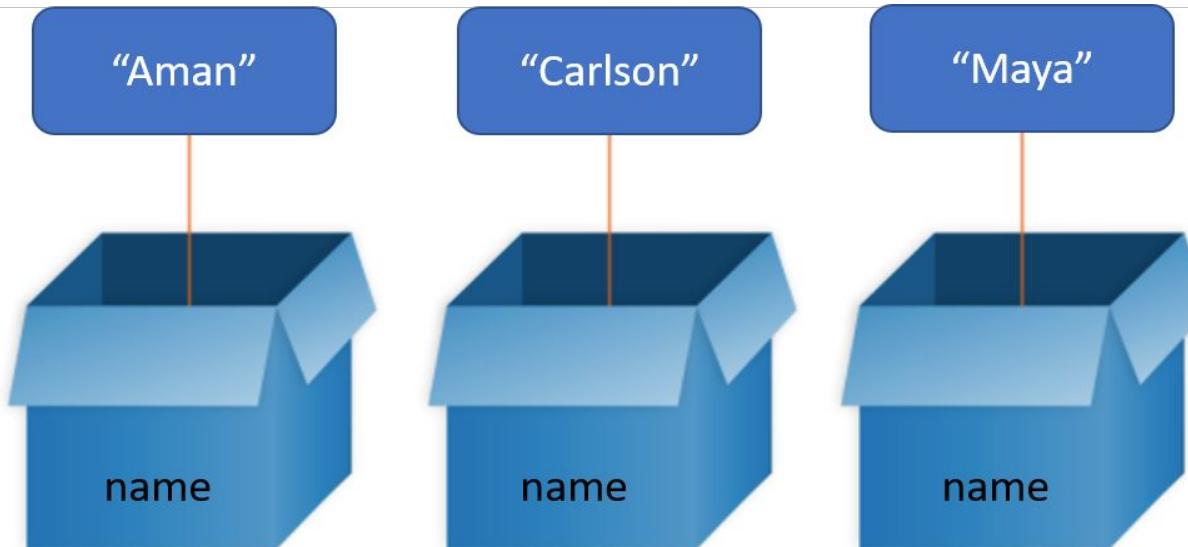




Variable Naming

- Name can begin with \$ and _ characters.
- You should not use any of the JavaScript reserved keywords as a variable name.
- JavaScript variable names should not start with a numeral (0-9).
 - For example, 123test is an invalid variable name but _123test is a valid one.
- JavaScript variable names are case-sensitive.

Need of Variable



var custName;

custName = "Aman"; or custName = "Maya"

Storing customer's Age

The site may want to know customer's age as well.

```
var custAge = 28;
```

However, *custAge* stores a **number** unlike *custName*
which stores a word (or *string*, as its called in JS)

Data types



Variables are like containers,

but different containers hold different Types of things

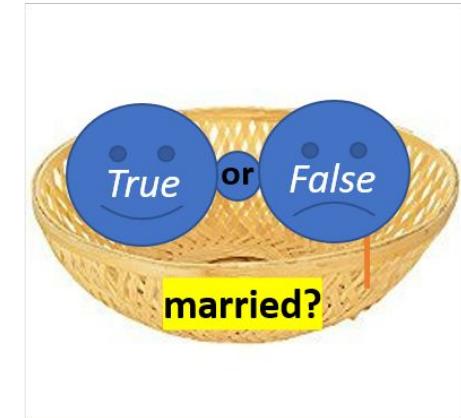


DATA
TYPES

NUMBER



STRING



BOOLEAN
(True or False)

Variables are like containers,

but different containers hold different Types of things

Data Type

Defines the type of data
that a variable can hold



This page says

What's your name?

Cancel

OK



Data Type

- Number
- String

```
var age = 10; //number data type - the numerical value 10 is important here
```

```
var name = "masai"; //string data type
```

String

Aa Bb Cc Dd Ee

Ff Gg Hh Ii Jj

Kk Ll Mm Nn Oo

Pp Qq Rr Ss Tt

Uu Vv Ww Xx Yy Zz

1 2 3 4 5 + - × ÷ = ? &

6 7 8 9 0 × # < > ! @

The Cat in the Hat

These symbols can be put together to make words or phrases. Like “The Cat in the Hat”.

Remember all the symbols you learned in your
kindergarten class?

String



For the computer, it is just a group of symbols put together in a string.

String

Note: The symbols 0,1,2,3,4,5,6,7,8,9 can also be used inside a name as a string where they don't mean to show quantity.

In this Car Number Plate, the number symbols don't mean quantity!



Can you think of any other examples of this?

typeof

- Tells about data type of the variable.

Variable Exercise

- Name :
- Age :
- Favourite Food :
- Hobby :
- School Name :
- Shoe-Size:



Review and Feedback

[Start Poll](#)

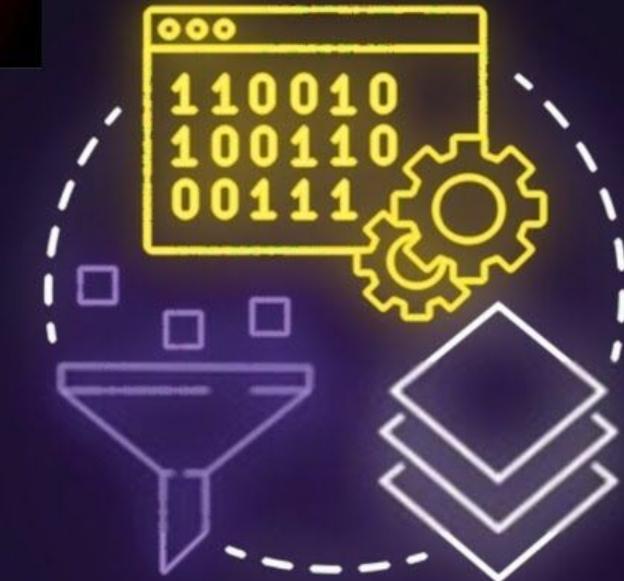
Interpreter

vs

Compiler



A B C D E F G H I J K L M N O P Q R
S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9
*/>< & # N°? ! @ % = + - \$ € £ (.,)



A B C D E F G H I J K L M N O P Q R
S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9
*/>< & # N°? ! @ % = + - \$ € £ (.,)

Variables

- Holding data

