Introduction to HTML and JavaScript

Introduction to HTML and JavaScript

Your goals for this course:

- Understand how the World Wide Web works: frontend, backend and their relationship
- Understand what a web designer does: role, tools, expectations
- Understand what a web developer does: role, tools, expectations
- Learn to make a basic website: create pages with HTML and add interaction with JavaScript

- Analyzing the web: overview
- Analyzing the web: front-end
- Analyzing the web: back-end
- Who is the web designer?
- Who is the web developer?
- HTML
- CSS
- JavaScript

Analyzing the web

overview

What is the **World Wide Web** (**WWW**)?

- It’s a network of **online content** and **web resources**.
- Every page of these contents is identified by a **URL**: **U**niform **R**esource **L**ocator.
- These pages are interlinked by **hypertext** links.
- All these pages can be accessed through **Internet**.
- In particular, they can be accessed through the **HTTP protocol** (**H**yper**T**ext **T**ransfer **P**rotocol).

- The **WWW** has been invented by **Tim Berners-Lee** in **1989**, at the **CERN** near Geneva, Switzerland.

What do we need to access the web?

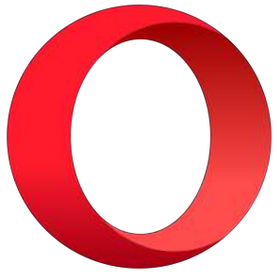
- An *internet connection*.

- A **Web Browser**: a software application which role is to access the World Wide Web.

The most used web browsers:




But there are others as well…

…and tons of others more.

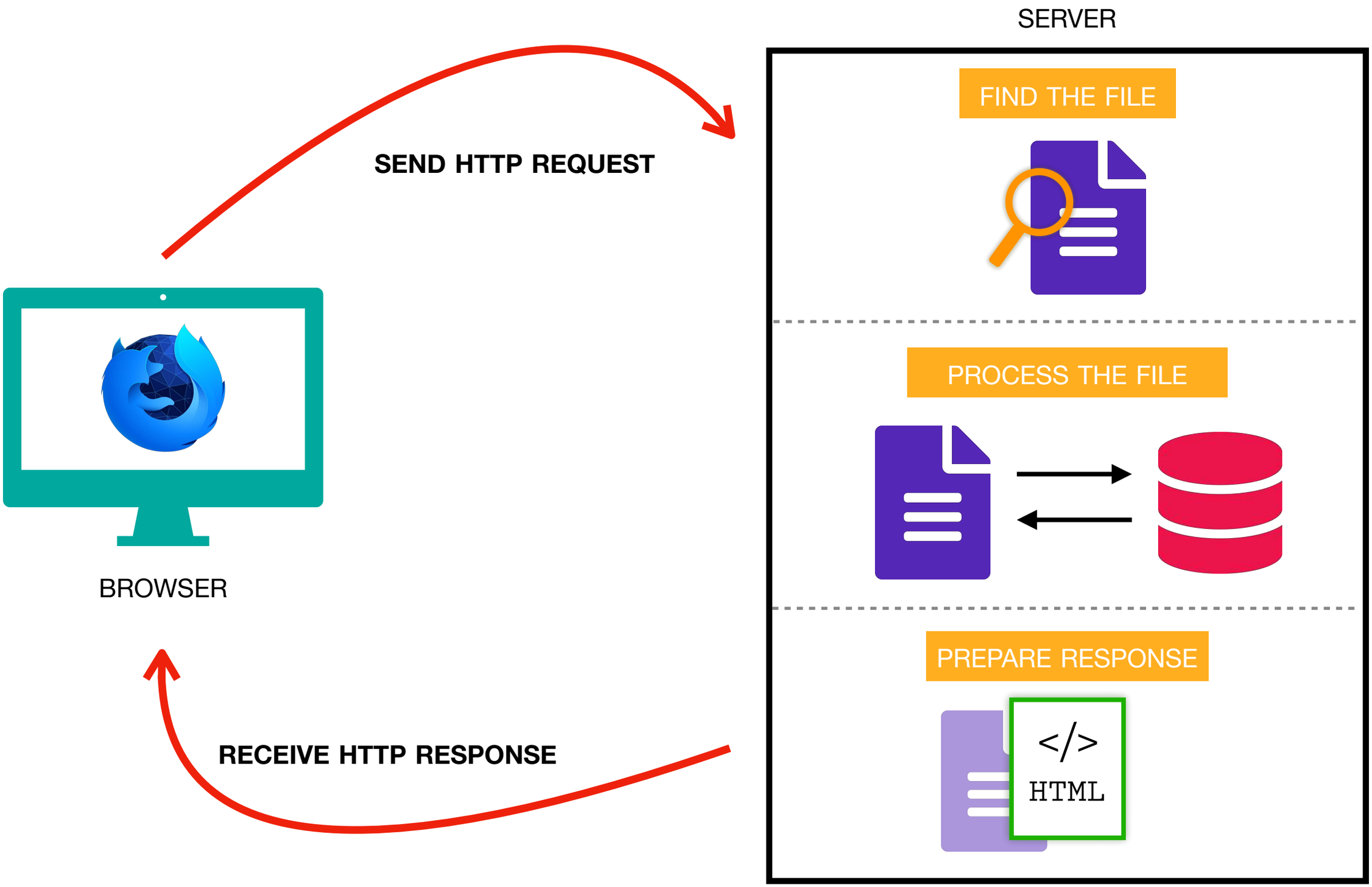
The best option for a web designer/developer:

FIREFOX DEVELOPER EDITION

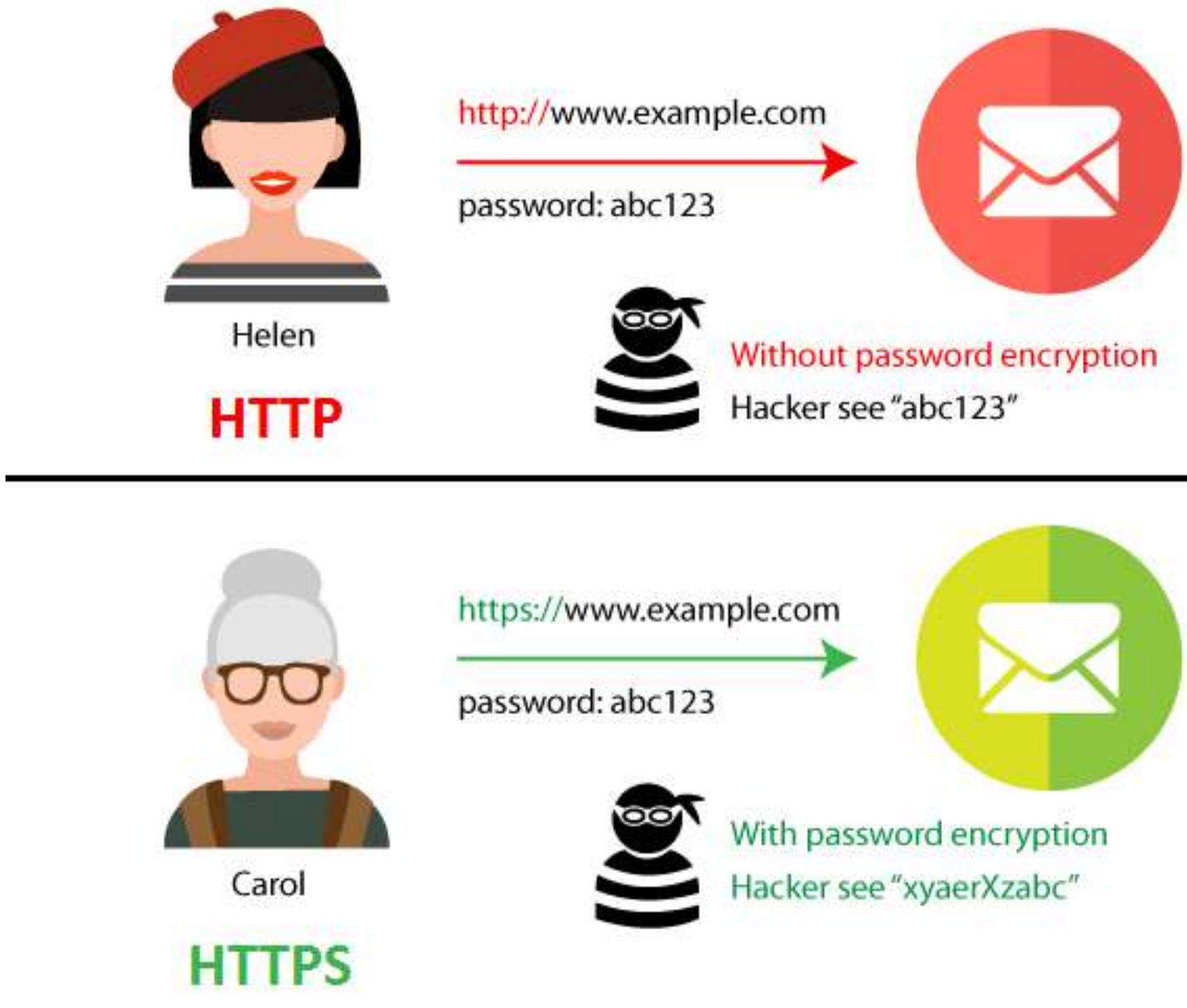
UNDERSTANDING HOW THE WEB WORKS

What happens when I open a website?

UNDERSTANDING HOW THE WEB WORKS



- Why some pages are **HTTP** and others are **HTTPS**?



UNDERSTANDING HOW THE WEB WORKS

- Each page we visit on the web is a **hypertext document**.
- Each page is composed in **HTML**.
- It looks beautiful (*maybe not!*) thanks to a language called **CSS**.
- It's interactive (or not) thanks to a language called **JavaScript.**
- Usually, if you type `www.example.com` without specifying a particular page, the server itself will look for the **index** file page (e.g. *index.html*).

UNDERSTANDING HOW THE WEB WORKS

- Each **webpage** can be **static** or **dynamic**:

STATIC WEBPAGE

Content **never** changes or changes **rarely**.

EXAMPLES

A portfolio website.

A product/company website.

Developers have to **manually change it** updating the code.

DYNAMIC WEBPAGE

Content **constantly** changes or changes **often**.

EXAMPLES

A blog.

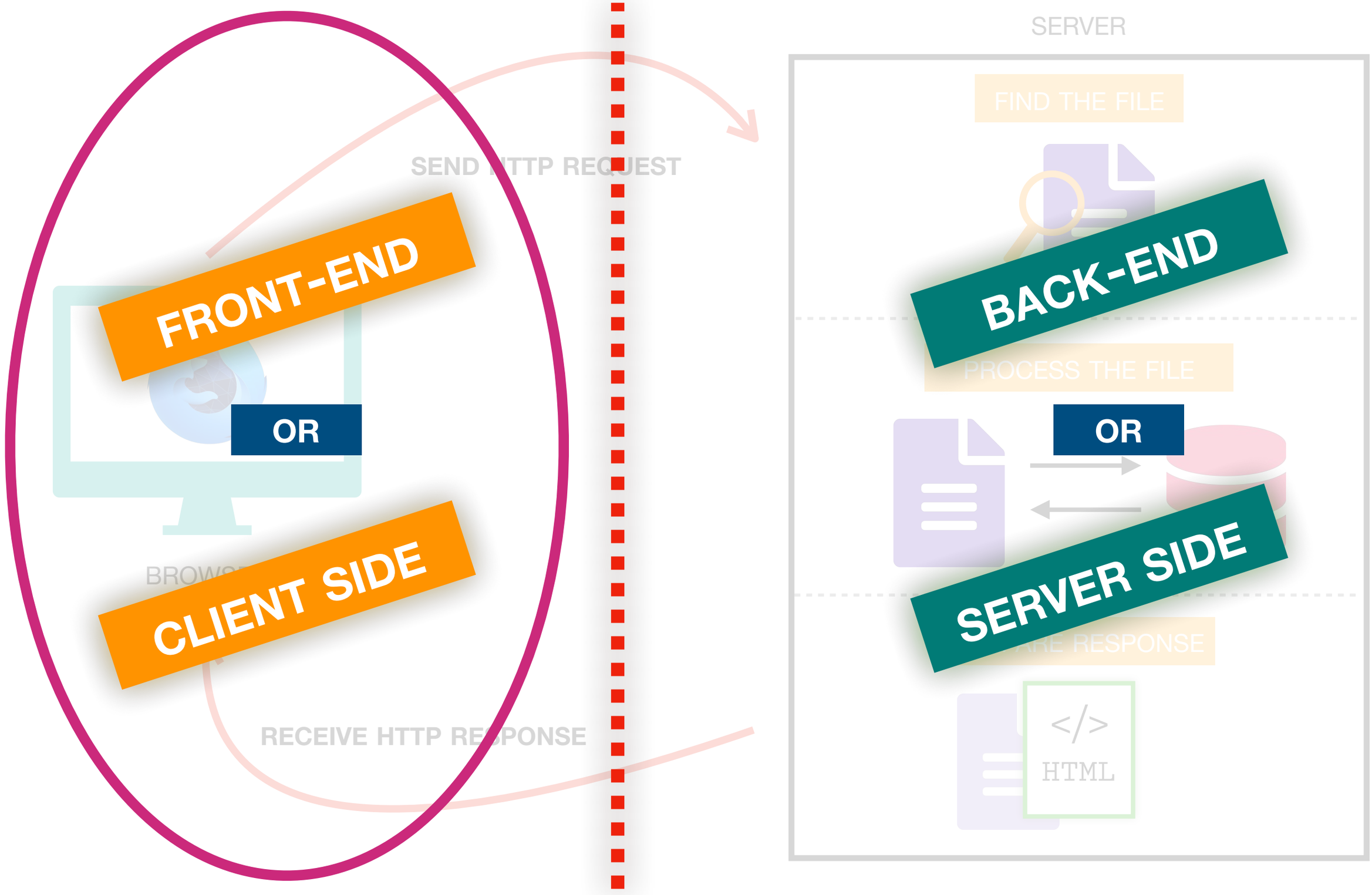
A social network.

Developers **don't need** to manually change it - the content is **updated automatically**.

- Also, pages are really often built **on-the-fly**.

Analyzing the web front-end

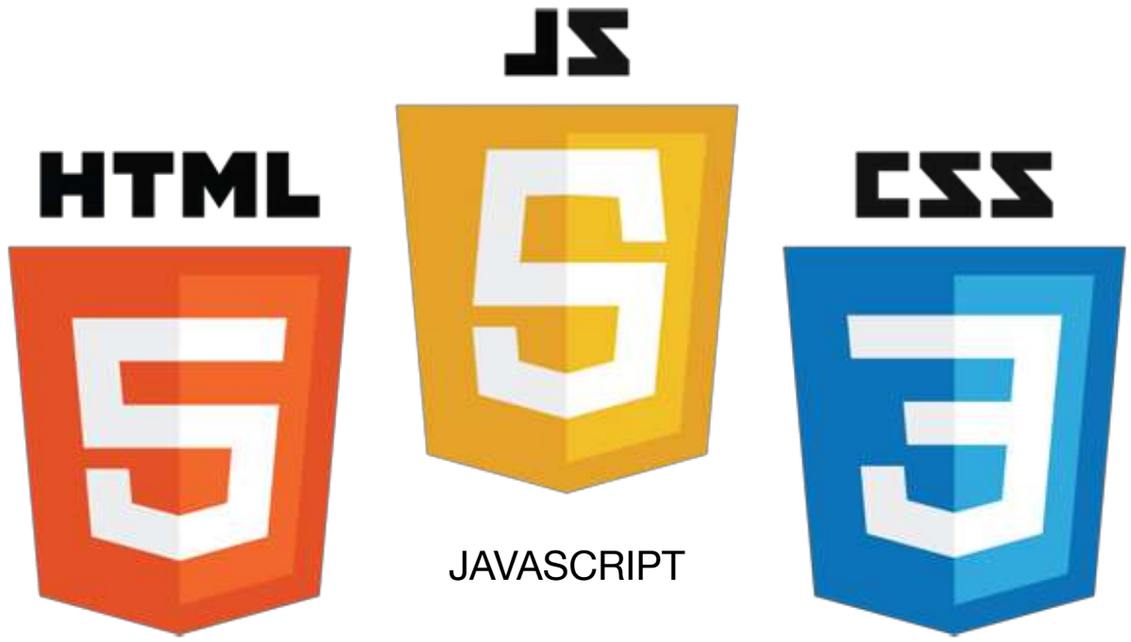

FRONT-END
BACK-END


SERVER
FRONT-END
SEND HTTP REQUEST
FIND THE FILE
BACK-END
PROCESS THE FILE
OR
OR
CLIENT SIDE
SERVER SIDE
RECEIVE HTTP RESPONSE
</>
HTML

UNDERSTANDING THE FRONT-END

- Front-end or **client side** is **what we see** displayed in our browser when we open a website.

- Front-end languages used by web designers are:



HTML CSS

UNDERSTANDING THE FRONT-END

- The purpose of front-end is to *structure* and *display* the data that forms a webpage.

- In this way, the user can see and interact directly with the website.

- In the front-end, so, it’s essential to consider not only how to make the webpage look like, but the **user experience** as well.

- This translation from code (*html*, *css*, *javascript*) to a graphical interface is made directly by the browser.

UNDERSTANDING THE FRONT-END

THIS (HTML)….

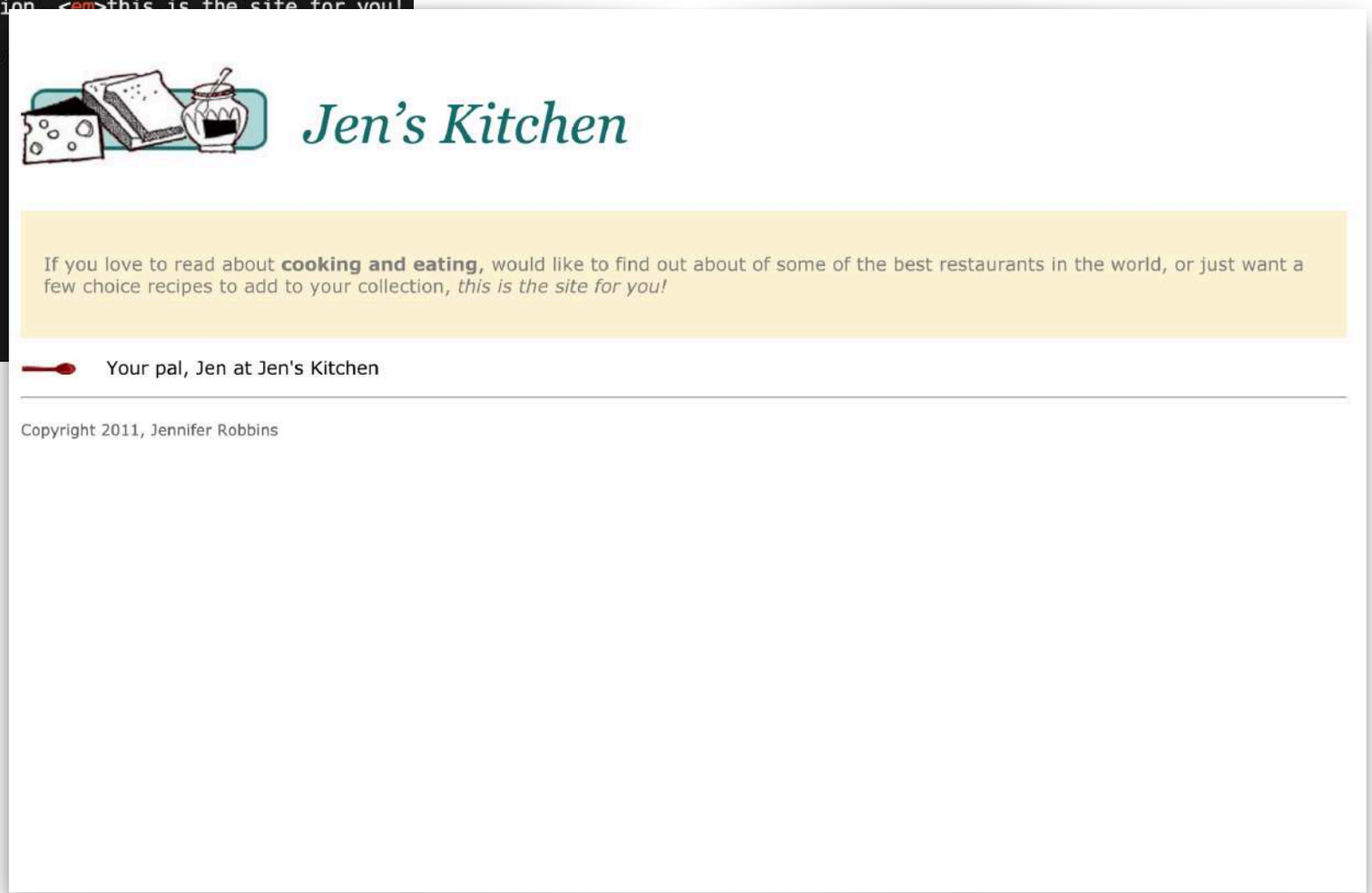
```
<!DOCTYPE html>
<html>

<head>
    <title>Jen's Kitchen</title>
    <link rel="stylesheet" href="style/kitchen.css" type="text/css">
</head>

<body>
    <h1><img src="images/foods.jpg" alt="food illustration"> Jen&rsquo;s Kitchen</h1>
    <div class="content"><p>If you love to read about <strong>cooking and eating</strong>,
    would like to find out about of some of the best restaurants in the world,
    or just want a few choice recipes to add to your collection, <em>this is the site for you!
    </em></p></div>
    <p><img src="images/spoon.jpg" alt="spoon illustration">
    >
    <hr>
    <p><small>Copyright 2011, Jennifer Robbins</small></p>
</body>

</html>
```

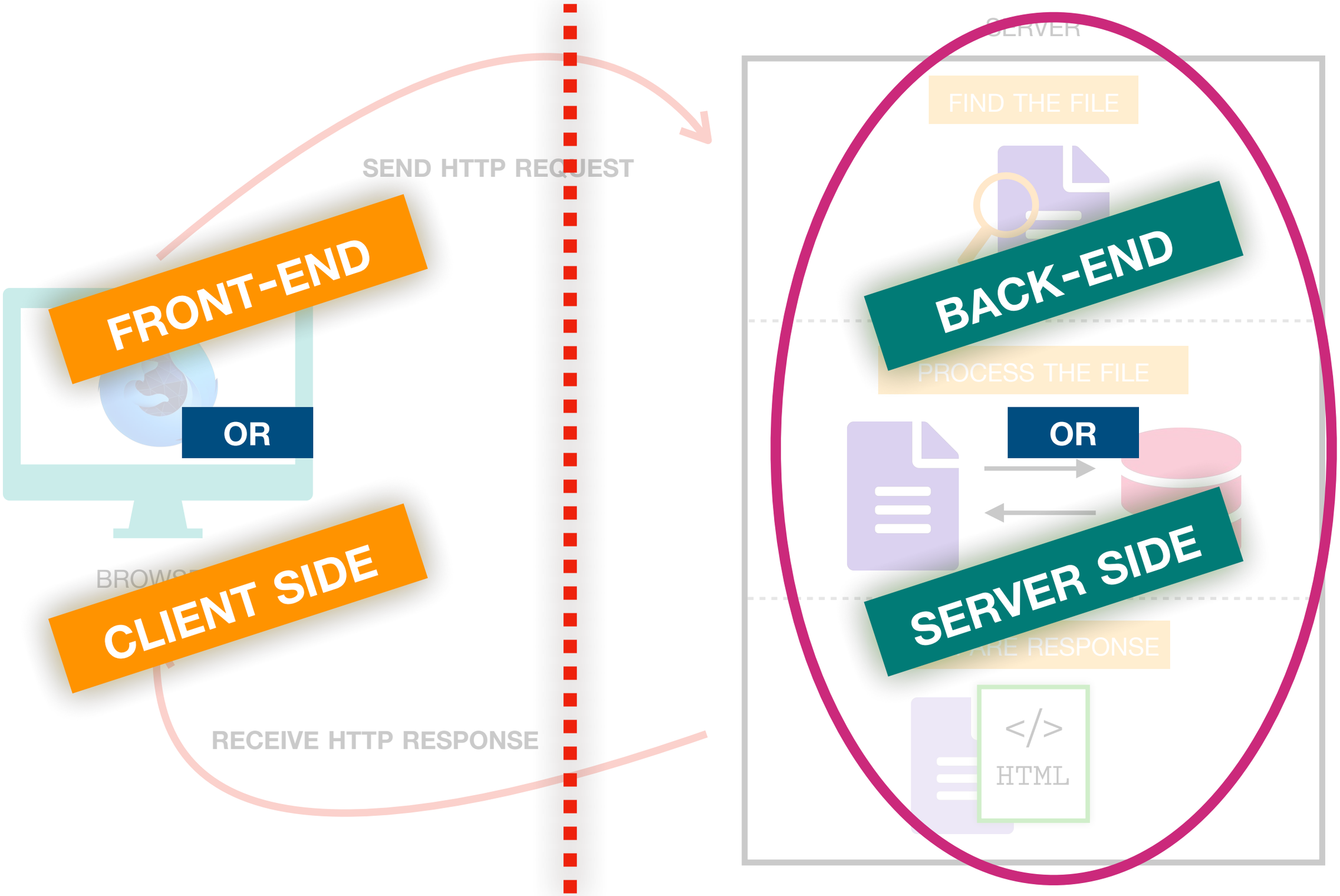
…CORRESPONDS TO THIS.



FRONT-END GOALS

- **Structure** the data in a webpage.
- Give the webpage a **style**.
- Give the webpage **interaction**.
- Create a good **user experience**.

Analyzing the web back-end


SEND HTTP REQUEST
FRONT-END
OR
CLIENT SIDE
RECEIVE HTTP RESPONSE
SERVER
FIND THE FILE
BACK-END
PROCESS THE FILE
OR
SERVER SIDE
RE RESPONSE
</>
HTML

UNDERSTANDING THE BACK-END

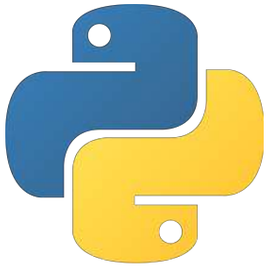
- Back-end or **server side** is **what we don't see** because it's being performed on the server.

- Back-end languages used by web developers can be:

RUBY

PERL

PHP



PYTHON

- …and others more, like Java, ASP.NET, **SQL**.

UNDERSTANDING THE BACK-END

What is a database?

- A collection of organised information that can be easily **accessed**, **managed** and **updated**.

- Essentially, a database is where all the data and information are written and read.

- For example, when you register on a website all your given information (username, password, email, etc.) are stored in a database.

UNDERSTANDING THE BACK-END

- The purpose of back-end is to make the website work correctly.

- A user can **NOT** see anything of the server side: not even which back-end programming language has been used.

- Imagine like a restaurant: you come and sit in the dining room (*front-end*), while the food is prepared in the kitchen (*back-end*) where you are not supposed to go. The ingredients, instead, are kept in the fridge (*database*).

- Plus, a back-end web developer has to make the website **secure**.

BACK-END GOALS

- Make the website **work correctly**.
- Make the website **secure** from *crackers*.
- **Maintain the information** in the database.

Who is the web designer?

THE ROLE

- As we have seen, the role of the web designer is to create a simple and beautiful **customer experience**.

- Also, a good web design is a design that follows the **identity** of the company which represent - that's why design skills are essential to perform this role:

- For example, a *bank website* can not have the same look and feeling of a *snowboard brand website*: in the first case, you may want to look professional and reliable - while in the second you may want something fresh, dynamic, youthful.

THE ROLE

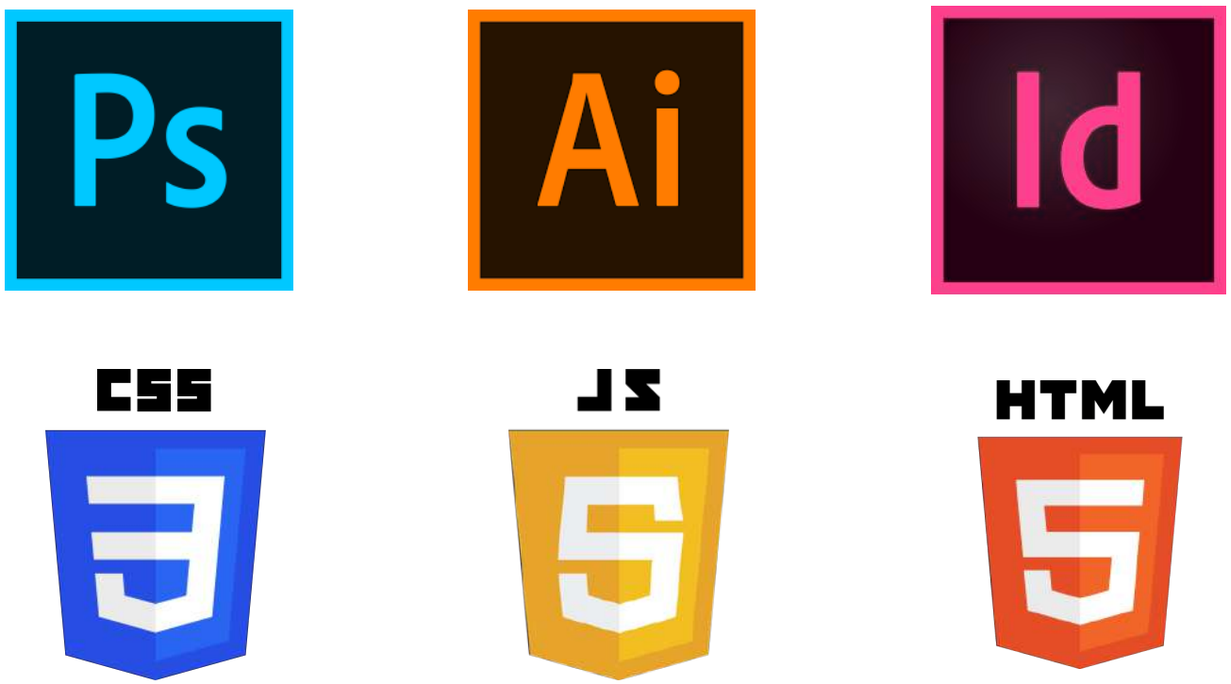
- As a web designer you have to mix together your knowledge of **visual communication** with more **logical** knowledge in terms of coding.

- If the website has already been made, the role of the web designer is to maintain and update its look and features.

- The web designer usually work in a team.

THE TOOLS

- Being that a web designer mixes design skills with coding skills, the tools used vary from **graphic software** to **programming languages**:



- The graphic software are used to create *mockups*, *images*, *icons*, while the programming languages to give life to the web design.

THE EXPECTATIONS

- Take care of the **look**, **layout** and **features** of a website.
- Understand the **brand** of the company and give the website the right **mood**.
- Understand the **needs** of the customer and fulfill them.
- Create something that is **trendy** and looks **cutting edge**.
- Create a good **responsive** web design: a design which adapt itself to the device used to connect to the website, making navigation easier and comfortable.
- Sometimes also **writing**, **editing** or **creating** contents.
- Be sure the web design works correctly on **different devices and browsers**.

Who is the web developer?

THE ROLE

- The role of the web developer, instead, is to *give life* to a website and make sure that everything works **correctly**.

- As web developer, so, your role is to create a **functional environment** which responds to all the customer's needs.

- Also, the ability of creating and maintaining an optimal **database** is essential.

THE ROLE

- **Logical** skills and **technical knowledge** in terms of coding are essential to fulfill this role at its best.

- The web developer has to take care of the **security** of the website as well: for example, processing the user inputs before proceeding with anything else.

- The rule #1 of a web developer is: **never trust users**.

THE TOOLS

- Web developers tests their applications using **local virtual servers**. Also It's common nowadays for web developers to build their applications upon **frameworks**:

- The graphic software are used to create *mockups*, *images*, *icons*, while the programming languages to give life to the web design.

THE EXPECTATIONS

- Take care of the **correct operation of a website**.
- Create, keep and maintain the **database**.
- **Manage** the data that the user provides to the website.
- Create a **secure** environment protected as much as possible from malicious hacking attacks.
- Give life to a website.

HTML

WHAT IS HTML?

- **HTML** stands for **H**yper**T**ext **M**arkup **L**anguage.
- It's been designed by **Tim Berners-Lee** during the 90s.
- It's the standard *markup language* used for creating webpages.
- Its role is to *describe* the structure of the document: it tells the browser how to display text and images.
- For example, with HTML you can define paragraphs of text, titles, underline text, insert images, etc.

```
<!DOCTYPE html>
<html>

<head>
    <title>Jen's Kitchen</title>
    <link rel="stylesheet" href="style/kitchen.css" type="text/css">
</head>

<body>
    <h1><img src="images/foods.jpg" alt="food illustration"> Jen&rsquo;s Kitchen</h1>
    <div class="content"><p>If you love to read about <strong>cooking and eating</strong>,
would like to find out about of some of the best restaurants in the world,
    or just want a few choice recipes to add to your collection, <em>this is the site for
you!</em></p></div>
    <p><img src="images/spoon.jpg" alt="spoon illustration"> Your pal, Jen at Jen's
Kitchen</p>
    <hr>
    <p><small>Copyright 2011, Jennifer Robbins</small></p>
</body>


</html>
```

CSS

WHAT IS CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets.
- It's been designed by **Hakon Wium Lie** during the 90s while working with Tim Berners-Lee.
- It's the standard *style sheet language* used for formatting webpages.
- Its role is to *give style* to the document: colors, position, fonts, etc. - it also define the **responsiveness** of the document.
- It can be used in three ways:
- **INLINE**: the style is defined directly inside the *tag* of the element
- **INTERNAL**: the style is defined directly inside the html file between <*style*> tags.
- **EXTERNAL**: the style is defined in an external file with extension *.css* and linked with the HTML file.

INLINE CSS

```
<!DOCTYPE html>
<html>
<head>
    <title>Jen's Kitchen</title>
</head>
<body>
    <p style=“color:blue;font-size:18pt;”>Your pal, Jen at Jen's Kitchen</p>
</body>
</html>
```

INTERNAL CSS

```
<!DOCTYPE html>
<html>
<head>
    <title>Jen's Kitchen</title>
    <style>
      p{
        color:blue;
        font-size:18pt;
      }
    </style>
</head>
<body>
    <p>Your pal, Jen at Jen's Kitchen</p>
</body>
</html>
```

INTERNAL CSS

```
<!DOCTYPE html>
<html>
<head>
    <title>Jen's Kitchen</title>
    <link rel="stylesheet" href="style/kitchen.css" type="text/css">
</head>
<body>
    <p>Your pal, Jen at Jen's Kitchen</p>
</body>
</html>
```

CSS

```css
body {
    font: normal 1em Verdana;
    margin: 1em 10%;
    background-image:url(images);
}

h1 {
    font: italic 3em Georgia;
    color: rgb(23, 109, 109);
    margin: 1em 0 1em;
}

img {
    margin: 0 20px 0 0;
}

h1 img {
    margin-bottom: -20px;
}

small {
    color: #666666;
}

.content {
    padding:20px;
    background-color:#FCE9A9;
    opacity: 0.5;
}
```

JavaScript

WHAT IS JAVASCRIPT?

- **JavaScript** also known as **JS**, has been invented by **Brendan Eich** in 1995.

- Together with HTML and CSS is one of the core technologies of the web.

- Its role is to add *interaction* to the webpage: it's used to manipulate and process data, which is taken from and written into the **DOM** (**D**ocument **O**bject **M**odel). The elements inside the HTML are called **DOM nodes**.

- JavaScript is an extremely powerful *script language.*

- Javascript can be written directly inside the HTML file (in particular inside the <*head*> tags or at the very end of the <*body*> tags) and it's identified by the tags <*script*> or in an external file with extension *.js*

```
<script>
  function changeColor(){
     var x=document.getElementById("change");
     var y=x.getElementsByClassName("child");
     var i=0;
     while (i < y.length){
        y[i].style.backgroundColor="green";
        i=i+1;
        }
     }
</script>
```

That's all Folks!