

# CSE391 - Nền tảng phát triển Web

## Giới thiệu môn học

dungkt@tlu.edu.vn

# Nội dung

- 1) Thông tin giảng viên
- 2) Mô tả khóa học
- 3) Mục tiêu và kết quả
- 4) Đánh giá môn học
- 5) Tài liệu tham khảo
- 6) Một số qui định của môn học

# 1. Thông tin giảng viên

- Kiều Tuấn Dũng
- Phòng làm việc: Bộ môn HTTT, Tầng 2, Nhà C1
- Email: [dungkt@tlu.edu.vn](mailto:dungkt@tlu.edu.vn)
  - Đề nghị khi trao đổi qua email, gửi mail với cấu trúc TIÊU ĐỀ THƯ [CSE391] Em hỏi về thẻ HTML5 (Ví dụ)
  - Thầy sẽ không check các email thực hiện không đúng yêu cầu.
- Mobile: 0868.600.513
  - Nếu gọi không được, vui lòng để lại tin nhắn, thầy sẽ gọi lại khi phù hợp.

## 2. Mô tả khóa học

- Trong khóa học này, chúng ta sẽ trải qua các kiến thức bao gồm:
  - Quản lý source code và làm việc nhóm với GIT.
  - Quy trình phát triển website và các ứng dụng dựa trên nền Web
  - Phát triển Web phía Front End: Thiết kế website với HTML, CSS, JavaScript , Jquery, Responsive, Bootstrap, dàn layout từ Photoshop
- Lớp học sẽ bao gồm các Buổi học Lý thuyết + Thực hành (**Đề nghị sinh viên chủ động mang Laptop để qua trình học có thể diễn ra liên tục, cả trên phòng học Lý thuyết + Thực hành**)
- Khóa học bao gồm 4 bài tập nhỏ + 1 dự án cuối môn học [ Nhóm 2 người].
- Khóa học sử dụng 2 kênh giao tiếp chính với sinh viên: **Facebook Group: để thông báo nhanh và các trao đổi khác & Website: hoccong nghethongtin.com để lấy tài liệu và nộp bài tập.**



### 3. Mục tiêu và kết quả môn học

- Mục tiêu chung: Giới thiệu cho sinh viên các khái niệm, kỹ thuật, hệ thống và thực hành hiệu quả để phát triển các ứng dụng trên nền web. Mục tiêu cụ thể bao gồm:
  - Hiểu biết về ngôn ngữ đánh dấu, ngôn ngữ định kiểu trang web, về mô hình lập trình phía máy khách và phía máy chủ cho các ứng dụng web.
  - Phát triển các kỹ năng cần thiết cần thiết cho lập trình web.
  - Trải nghiệm và tìm hiểu về sự phát triển của các ứng dụng web.
- Cuối khóa học, sinh viên có thể:
  - Giải thích các khái niệm và công nghệ cơ bản trong World Wide Web.
  - Thiết kế và triển khai các giao diện web bằng HTML, CSS và JavaScript, jQuery, Bootstrap
  - Chuyển tệp PSD sang HTML, CSS



## 4. Đánh giá môn học

Hình thức	Số lần	Mô tả	Thời gian	Trọng số
<b>1. ĐIỂM QUÁ TRÌNH</b>				40%
Chuyên cần	1	Đi học đầy đủ Thái độ học tập tích cực	- Xuyên suốt môn học	20%
Bài tập #1	1	Bài tập nhỏ trên lớp (HTML, CSS)	- Tuần 2	20%
Bài tập #2	1	Bài tập nhỏ trên lớp (Javascript)	- Tuần 4	20%
Bài tập #3	1	Bài tập nhỏ trên lớp (jQuery)	- Tuần 5	20%
Bài tập #4	1	Bài tập nhỏ trên lớp (Bootstrap)	- Tuần 6	20%
<b>2. ĐIỂM THI KẾT THÚC HỌC PHẦN</b>				60%

**Bài tập để đánh giá sẽ xem xét  
dựa trên sự kết hợp của 3 tiêu  
chí ASK:**

1. Thái độ
2. Kỹ năng
3. Kiến thức

## 5. Tài liệu tham khảo

- Đề cương chi tiết, Slide Bài giảng & Bài tập: được cung cấp tại <http://hoccongnghethongtin.com>
- Video Bài giảng (Youtube)
- Giáo trình [**Khuyến khích đọc Giáo trình tiếng Anh**]:
  - Zak Ruvalcaba, Anne Boehm, **Murach's HTML5 and CSS3, 4th Edition, Mike Murach & Associates**, 2018
  - Randy Connolly, **Fundamentals of Web Development**, Pearson, 2018
- Website tham khảo:
  - <https://www.w3schools.com/>
- **Copyright:** Nội dung được tham khảo từ devpro, có chỉnh sửa và cập nhật.

## 6. Một số qui định của môn học

- Sinh viên đi học muộn quá **5 phút** so với giờ vào tiết học, vui lòng chờ đến tiết sau vào lớp để không làm ảnh hưởng đến diễn tiến của lớp học.
- Sinh viên được nghỉ học theo qui định, qui chế đào tạo tối đa **20%** thời lượng môn học (dành cho các trường hợp ốm đau, gia đình có việc đột xuất)
  - Trừ trường hợp đặc biệt được xem xét cụ thể nếu nghỉ quá số buổi học (năm viện có giấy xác nhận, gia đình có việc hiếu ...)
  - Sinh viên không xin phép nghỉ với các lý do: em có việc bận, gia đình em có việc ...
- Sinh viên làm và nộp bài tập tự giác, đầy đủ đúng qui định.
- Sinh viên cài đặt đủ phần mềm theo yêu cầu của môn học và mang theo Laptop khi đến lớp.

# CSE391 - Nền tảng phát triển Web

## Bài 01: Giới thiệu

dungkt@tlu.edu.vn

# Nội dung



- 1) Giới thiệu
- 2) Quy trình phát triển Web
- 3) Các công cụ cần có
- 4) Cách học hiệu quả

# 1. Giới thiệu

# Web Development Roadmap



- **Tham khảo chi tiết:**
  - <https://hackernoon.com/the-2019-web-developer-roadmap-ab89ac3c380e>



# Web tĩnh và Web động

- **Web tĩnh**, nghĩa là khi trang Web được tạo ra và lưu lại dưới định dạng HTML, nội dung và cấu trúc của trang sẽ không thay đổi cho tới khi mã nguồn được chỉnh sửa.





# Web tĩnh và Web động

- **Web động**, nghĩa là khi trang Web được tạo ra và lưu lại dưới định dạng của ngôn ngữ lập trình Web động (ví dụ: **PHP**, ASP.net), nội dung và cấu trúc của trang sẽ thay đổi theo cách nó được lập trình và tương tác với người dùng + Cơ sở dữ liệu (**MySQL**, SQL Server...)

## Fullstack

SQL  
PHP  
ASP  
Python

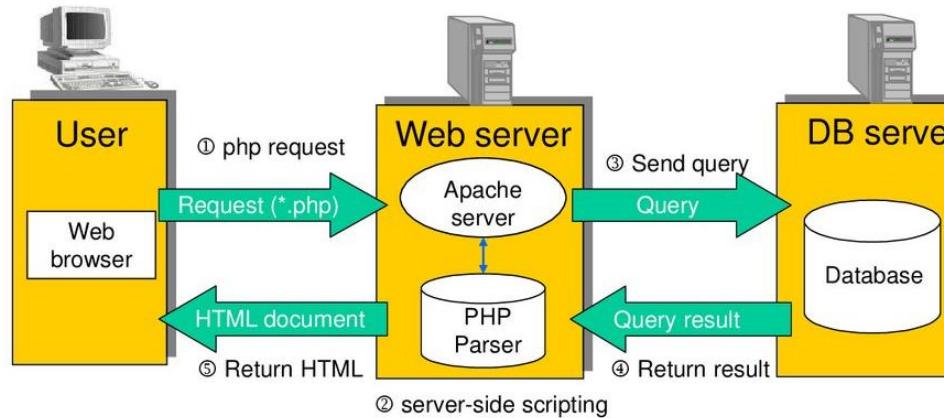
## Fullstack JS

SQL  
Node.js  
MySQL  
Mongo.db

# Web Server



- Thường là máy tính có cấu hình cao được kết nối ra ngoài Internet, hoạt động 24/24
- Cài đặt hệ điều hành máy chủ (Ubuntu Server, CentOS, Windows Server)
- Cài đặt các ứng dụng máy chủ Web (Apache, Nginx, IIS ...)
- Chứa mã nguồn ứng dụng Web/Website



# Tổng quan về Khóa học



- **CSE391:** 45 tiết
  - HTML - CSS - JS - jQquery - Bootstrap: 45 tiết
- **CSE485:** 45 tiết
  - PHP căn bản
  - Cơ sở dữ liệu MySQL - Kết nối MySQL với PHP
  - Các kỹ thuật nâng cao
  - Lập trình hướng đối tượng PHP - Mô hình MVC
- **CSE\*\*\*:**
  - Dựng trang bán hàng với MVC
  - Tìm hiểu Framework Laravel - Dựng trang bán hàng với Laravel

## 2. Quy trình phát triển web

# Quy trình phát triển Web thông thường



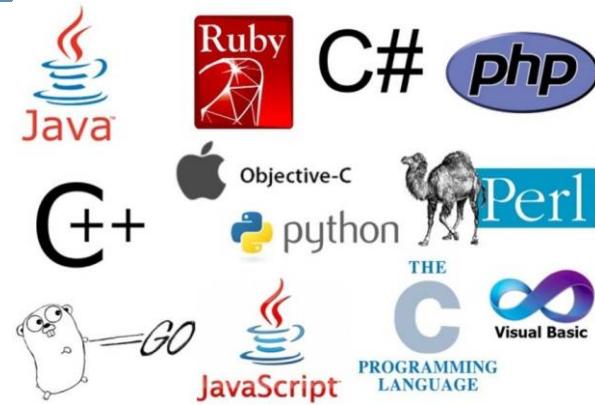


# Định hướng Lập trình

- Phát triển Frontend



- Phát triển Backend



- Phát triển Fullstack



- Top 10 Web CMS

1. [WordPress](#) (27+ Million Live Websites)
2. [Wix](#) (3.8+ Million Live Websites)
3. [Squarespace](#) (1.9+ Million Live Websites)
4. [Joomla!](#) (1.8+ Million Live Websites)
5. [Shopify](#) (1.1+ Million Live Websites)
6. [Drupal](#) (630+ Thousand Live Websites)
7. [Blogger](#) (430+ Thousand Live Websites)
8. [Prestashop](#) (285+ Thousand Live Websites)
9. [Magento](#) (265+ Thousand Live Websites)
10. [Bitrix](#) (223+ Thousand Live Websites)



# Đặc trưng của PHP

- Mã nguồn mở, chuyên dụng cho môi trường Web
- Tính cộng đồng, nhiều tài liệu hướng dẫn và hỗ trợ
- Thư viện phong phú
- Hỗ trợ nhiều Hệ CSDL
- Lập trình Hướng đối tượng
- Tính bảo mật
- Khả năng mở rộng không giới hạn

### 3. Công cụ cần có

# Text Editor/IDE



- Text Editor: Visual Studio Code + extension/add-on
  - Sublime Text, Atom, Vim ...
- IDE: PHP Designer, PHPStorm, ZendStudio



Visual Studio Code

# Trình duyệt Web



- **Firefox**

BEST BROWSERS FOR WEB DEVELOPMENT	PRICE	DEFAULT SEARCH ENGINE	LICENSE
92  Firefox Developer Edition	0	-	-
89  Mozilla Firefox	FREE	Google	OpenSource (MPL-2.0)
85  Google Chrome	FREE	Google	Proprietary
82  Chromium	-	Google	OpenSource (BSD)
Brave 	-	Qwant	OpenSource (mainly MPL 2.0)

- **Chrome**

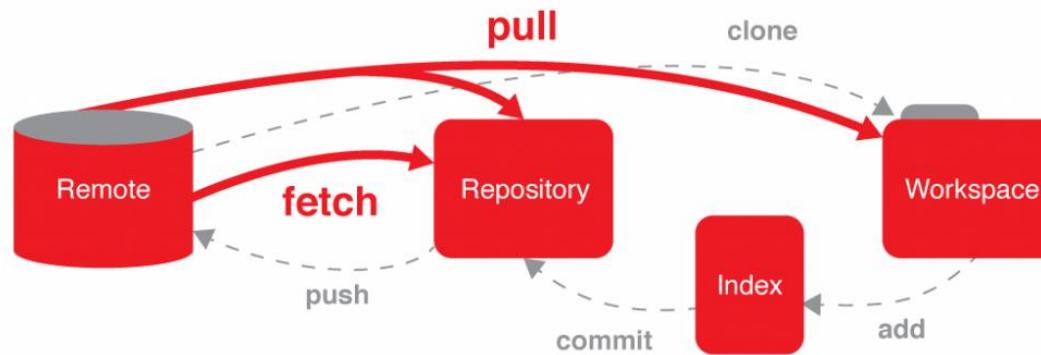
- **Edge**

- ...

# GIT



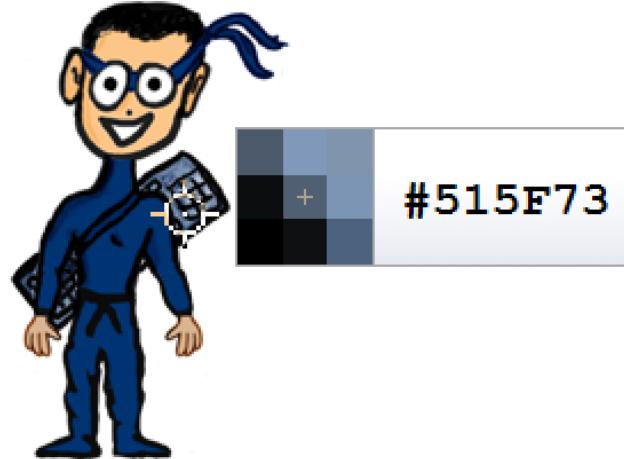
- Công cụ cần phải biết để làm việc nhóm sau này, mỗi buổi học của khóa cũng sẽ sử dụng GIT để lưu trữ & tài khoản (Github | Gitlab | Bitbucket)
  - Lập trình viên lĩnh vực nào cũng cần thành thạo và dung đến GIT



# Công cụ đồ họa



- **Instant Eyedropper:** Phần mềm nhỏ gọn cho phép lấy mã màu trên màn hình rất nhanh chóng mà không cần sử dụng Photoshop.
- **Photoshop:** Phần mềm chỉnh sửa ảnh, được sử dụng cho việc lấy mã màu, thiết kế logo, banner hoặc cắt HTML CSS khi cần.

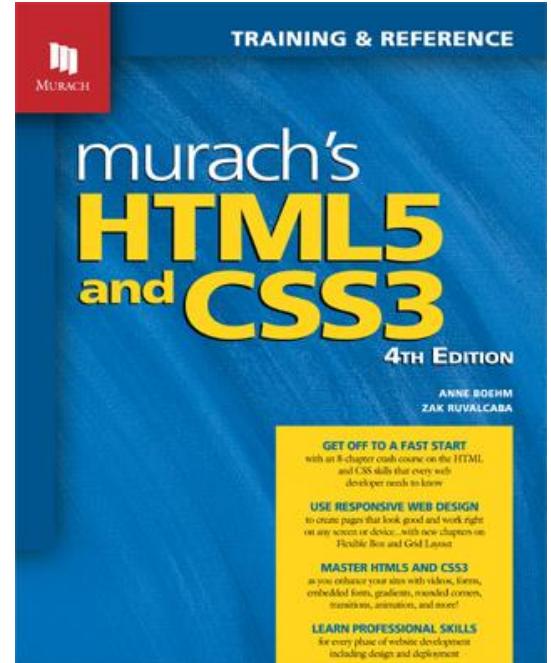
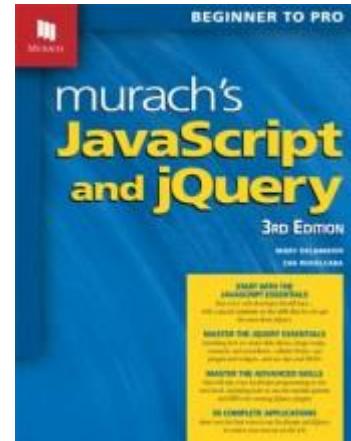


## 4. Cách học hiệu quả

# Học bằng sách



- Chọn 1 cuốn sách làm cẩm nang tra cứu lý thuyết (khi cần)
- Chọn 1 cuốn sách chứa Ví dụ thực hành theo một dự án thực tế
  - Có mã nguồn ví dụ kèm theo càng tốt (để học cách họ code)





# Học bằng Video

- Xem Video tới đâu, thực hành Code ngay tại đó.
  - Tránh ngồi xem cả chục Video ... vừa uống trà, vừa gật gù ... hay quá, dễ hiểu quá.
  - Một bên mở video, một bên mở Trình gõ code

HTML Crash Course For Absolute Beginners

2,924,038 views • Jul 17, 2017

71K 702 SHARE SAVE

HTML & CSS  
Traversy Media - 1 / 50

- 1 HTML Crash Course For Absolute Beginners
- 2 CSS Crash Course For Absolute Beginners
- 3 Build An HTML5 Website With A Responsive Layout
- 4 Flexbox CSS In 20 Minutes
- 5 CSS GRID LAYOUT
- 6 Build a Netflix Landing Page Clone with HTML, CSS & JS
- 7 Responsive Landing Page Using

Create Your Own Website

# Tài liệu tham khảo



- <https://www.w3schools.com/>

# CSE391 - Nền tảng phát triển Web

## Bài 2: HTML

dungkt@tlu.edu.vn

# Nội dung

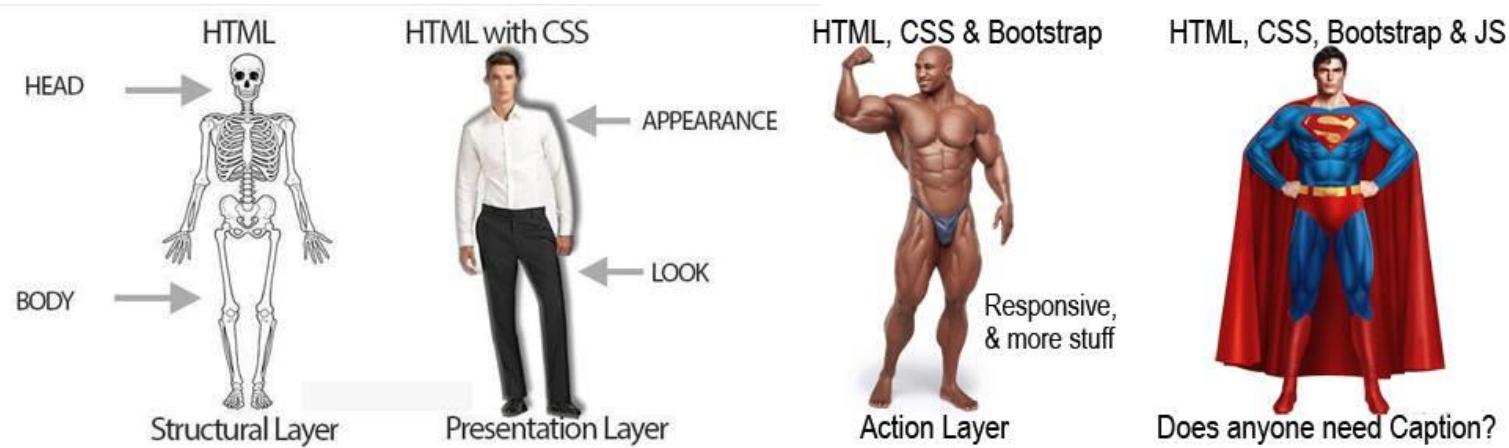


- 1) Khái niệm HTML
- 2) Cấu trúc trang HTML căn bản
- 3) Các thẻ HTML cơ bản
- 4) Các thẻ HTML5

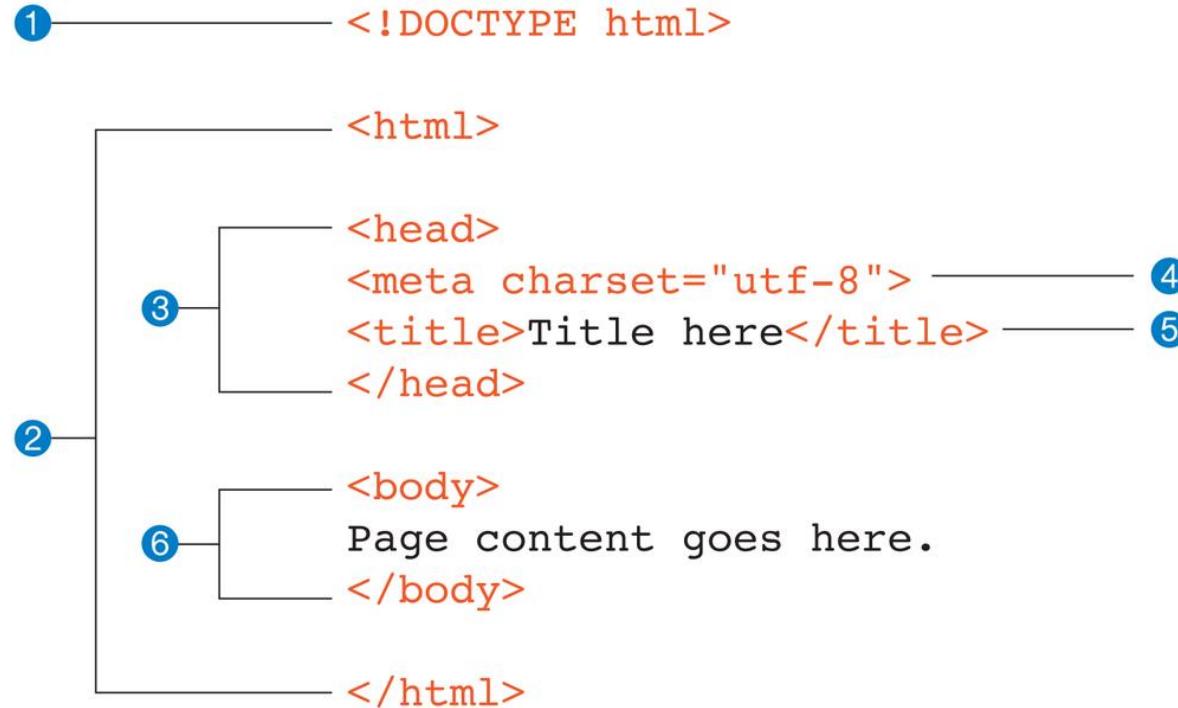


# Khái niệm HTML

- HTML - Hyper Text Markup Language
- Là ngôn ngữ sử dụng cho việc xây dựng bố cục cho trang web, thông qua các thẻ (tag)
- Thường kết hợp với CSS và Javascript để xây dựng nên giao diện hoàn chỉnh



# Cấu trúc trang HTML



# Các thẻ HTML



- <!DOCTYPE...> : Định nghĩa version HTML
- <html> : Bắt đầu trang html
- <head> : Bắt đầu phần header của trang HTML, chứa các thẻ mô tả cho header của trang như <title>, <link> .v.v..
- <title> : Tên trang web, sử dụng bên trong thẻ <head>
- <body> : Bắt đầu phần body của trang HTML, chứa các thẻ con khác, nội dung trong thẻ này sẽ được hiển thị trực quan tới client

# Các thẻ HTML



- <h1>, <h2>, <h3>, <h4>, <h5>, <h6> : Các thẻ heading, sử dụng để đặt tiêu đề cho các text cần nhấn mạnh, khác nhau về font chữ, <h1> font size lớn nhất, <h6> bé nhất
- <p> : Khai báo 1 đoạn văn bản
- <br /> : Nội dung sau thẻ này sẽ bị xuống dòng, lưu ý thẻ này không có thẻ đóng
- <hr /> : Tạo đường kẻ ngang, thẻ này không có thẻ đóng
- <i> : In nghiêng text bên trong thẻ

# Các thẻ HTML



- <b> : In đậm text bên trong thẻ
- <u> : Gạch chân text bên trong thẻ
- <div> : Thường sử dụng để nhóm các thẻ liên quan, dùng để chia bố cục cho trang Web
- <span> : Sử dụng bên trong thẻ khác để chỉnh sửa các phần tử theo ý muốn
- <meta> : Sử dụng trong thẻ <head>, định nghĩa các keyword của trang, thường sử dụng trong SEO. Các thuộc tính thẻ bao gồm: name, content, scheme, http-equiv

# Các thẻ HTML



- <img /> : Thêm hình ảnh vào trang web, không có thẻ đóng, bao gồm các thuộc tính:
  - src: đường dẫn tới file ảnh
  - title: hiển thị text khi hover qua ảnh
  - alt: hiển thị text khi ảnh không tồn tại
  - width: độ dài ảnh
  - height: chiều rộng ảnh
- <table>, <thead>, <th>, <tbody>, <tr>, <td>: Hiển thị bảng <table> với các hàng <tr> và các cột <td>
  - <thead> chứa <tr> và các <th> quy định phần header của bảng
  - <tbody> chứa các <tr> và <td> quy định phần body của bảng
  - Các thuộc tính của <table>: border, cellpadding, cellspacing
  - Các thuộc tính của <td>: rowspan, colspan

# Các thẻ HTML



- <ul>, <ol>, <li>: Các thẻ liệt kê
  - <ul> : sắp xếp không có thứ tự, có thuộc tính type nhận các giá trị square, disc, circle
  - <ol> : sắp xếp có thứ tự
  - <li> : tương ứng với 1 phần tử con trong <ul> và <ol>
- <a>: Thẻ liên kết dùng để điều hướng trên trang Web, có các thuộc tính:
  - Href : url điều hướng
  - Target : link được mở như thế nào, thông thường hay sử dụng giá trị \_blank - mở sang tab mới
- <button>: Tạo các nút có thể click được, có thể chứa text hoặc ảnh, đây là điểm khác biệt với thẻ <input />
- <center>: Căn giữa nội dung bên trong, khi học sang CSS thì sẽ không dùng thẻ này nữa



# Các thẻ HTML

- <form>, <input />, <textarea>, <select>, <option>
- <form>: tạo form HTML, có các thuộc tính sau:
  - Action : url submit form
  - Method : phương thức xử lý form (POST/GET)
  - Enctype : cách mã hóa dữ liệu, thường được khai báo khi upload file
- <input />: tạo các phần tử HTML, có các thuộc tính sau:
  - Type : kiểu input, được mô tả trong bảng ở slide sau
  - Name : dùng để xử lý trên server
  - Value : giá trị khởi tạo cho input

# CSE391 - Nền tảng phát triển Web

## Bài 3: CSS

dungkt@tlu.edu.vn

# Nội dung

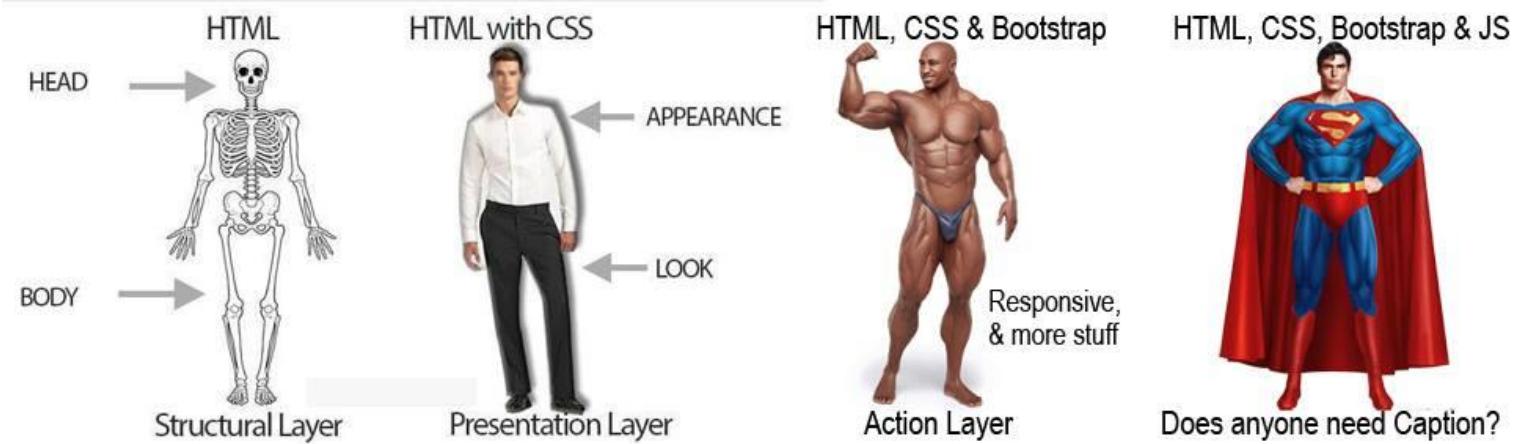


- 1) Khái niệm
- 2) Cú pháp khai báo
- 3) Selector
- 4) Nhúng file CSS
- 5) Thuộc tính CSS căn bản
- 6) Thành phần inline và block



# Khái niệm CSS

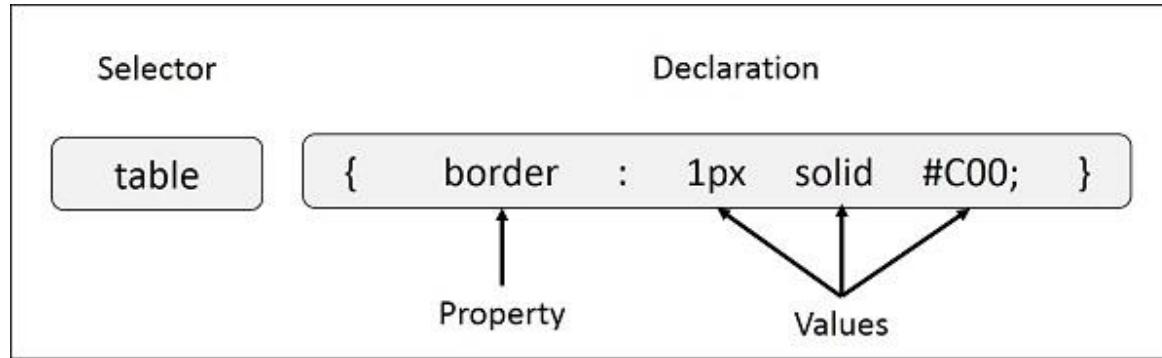
- CSS - Cascading Style Sheets
  - Ngôn ngữ hướng tới design Web bằng cách định dạng lại thẻ HTML





# Cú pháp khai báo

- selector { property: value }



- Ví dụ:
  - `table { border: 1px solid #c00; }`
  - `span { color: red; }`

# Selector: Bộ chọn



- Selector theo ID và Class
- Selector theo ID, sử dụng tiền tố #
  - #<id-name> {}
- Selector theo class, sử dụng tiền tố .
  - .<class-name> {}

# Selector: Bộ chọn



- Selector theo cấp
  - `#personal a {}`
  - `#personal > a {}`

# Selector: Bộ chọn



- Selector theo thuộc tính thẻ
  - `a[href='google.vn'] { color: red }`
  - `p[title='title'] { color: red }`

# Selector: Bộ chọn



- Selector theo trạng thái
  - :hover : Trạng thái khi hover qua selector. VD: a:hover { color: red;}
  - :first-child : Áp dụng cho phần tử đầu tiên trong mô hình cha con
  - :last-child : Áp dụng cho phần tử cuối cùng trong mô hình cha con
  - :before : Thêm phần tử vào trước selector, cần sử dụng thuộc tính content cho selector này
  - :after : Thêm phần tử vào sau selector, cần sử dụng thuộc tính content cho selector này



# Những tệp tin CSS

- Inline trong thẻ HTML

```
<a style="color: red;">A</a>
```

- Inline trong file HTML

```
<style type="text/css">  
    a {color: red};  
</style>
```

- External file

```
<link rel="stylesheet" href="/file.css" />
```

# Các thuộc tính CSS cơ bản



- Background: liên quan đến màu nền/ảnh nền
  - background-color
  - background-image
  - background-repeat
  - background-position
  - background
- Font: liên quan đến font chữ
  - font-family
  - font-style
  - font-weight
  - font-size

# Các thuộc tính CSS cơ bản



- Text: liên quan đến text
  - color
  - text-align
  - text-decoration
  - text-transform
  - text-shadow
  - line-height

# Các thuộc tính CSS cơ bản



- Border: liên quan đến viền của selector
  - border-color
  - border-style
  - border-width
  - border-radius
  - border

# Các thuộc tính CSS cơ bản



- Margin: khoảng cách giữa các phần tử HTML với nhau
  - margin-top
  - margin-bottom
  - margin-left
  - margin-right
  - margin

# Các thuộc tính CSS cơ bản



- Padding: khoảng cách giữa viền của phần tử đến nội dung bên trong nó
  - padding-top
  - padding-bottom
  - padding-left
  - padding-right
  - padding

# Các thuộc tính CSS cơ bản



- Width - Height: độ rộng - độ cao của phần tử
  - width
  - max-width
  - min-width
  - height
  - max-height
  - min-height

# Các thuộc tính CSS cơ bản



- Position: vị trí của phần tử so với các phần tử khác
  - position: relative
  - position: absolute
  - position: fixed

# Các thuộc tính CSS cơ bản



- Media: dùng trong responsive thiết bị
  - @media
- Float: dùng cho việc dàn trang
  - float: left
  - float: right
  - float: none
  - clear: both
- Overflow: cách hiển thị khi nội dung phần tử HTML bị tràn khỏi khung
  - overflow: visible
  - overflow: hidden
  - overflow: scroll

# Các thuộc tính CSS cơ bản



- Display: chế độ hiển thị của phần tử
  - inline
  - block
  - inline-block
  - none
  - table
  - table-cell
- Font-face: set font custom cho hệ thống
  - @font-face
- Z-index: độ ưu tiên hiển thị của phần tử so với các phần tử khác, chỉ có tác dụng với các phần tử đang có css position

# Thành phần inline và block



- Các phần tử HTML có thể chuyển đổi qua lại giữa 2 thành phần inline và block sử dụng thuộc tính CSS: display
- Phần tử inline: phần tử đứng liền kề nó không bị xuống dòng
- Phần tử block: phần tử đứng liền kề nó bị xuống dòng
- Thường sử dụng trong bố cục trang, tránh trường hợp sử dụng quá nhiều float dàn trang cho các phần tử đơn giản
- Một số thẻ inline: <b>, <i>, <u>, <a>, <span>
- Một số thẻ block: <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <div>

# CSE391 - Nền tảng phát triển Web

## Bài 4: PSD sang HTML và CSS

dungkt@tlu.edu.vn



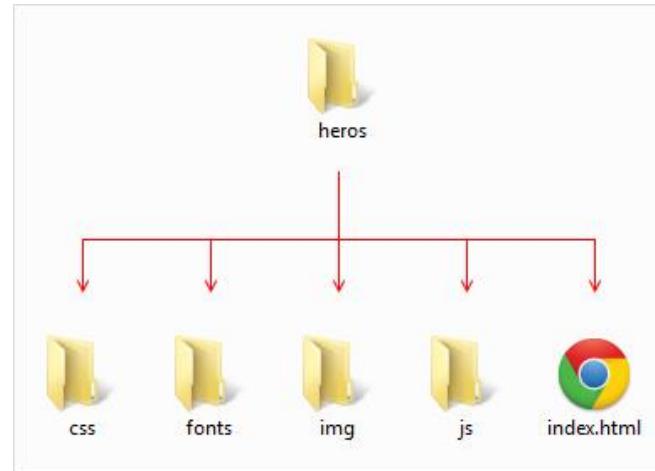
# Nội dung

- 1) Các bước thực hiện
- 2) Thực hành



# Các bước thực hiện

- Xây dựng cấu trúc thư mục lưu trữ
- Định nghĩa ra cấu trúc thư mục cần thiết cho việc cắt HTML CSS
- Cách đặt cấu trúc thư mục này là tùy chọn với mỗi cá nhân





# Các bước thực hiện

- Cắt ảnh từ file .psd
- Tiến hành cắt ảnh từ file .psd sử dụng công cụ Crop trong Photoshop
- Chỉ cắt các ảnh nào mà không thể dùng HTML CSS để tạo, để có thể dễ chỉnh sửa sau này





# Các bước thực hiện

- Lấy font của các text đang sử dụng từ file .psd
  - Sử dụng công cụ Text (phím tắt T) để lấy tên font chữ từ file .psd
  - Sau khi có được tên font, google search tên font và download các font này về máy, lưu vào thư mục fonts đã tạo trong cấu trúc thư mục ban đầu
  - Tiến hành import font vừa download về vào file .css theo mẫu sau, với giả sử file font PlayfairDisplay-Black.oft là font bạn vừa tải về

```
@font-face {  
    font-family: 'PlayfairDisplay-Black';  
    src: url("../fonts/PlayfairDisplay-Black.oft");  
}  
  
/*set font cho phần tử HTML*/  
body{  
    font-family: 'PlayfairDisplay-Black', sans-serif;  
}
```



# Các bước thực hiện

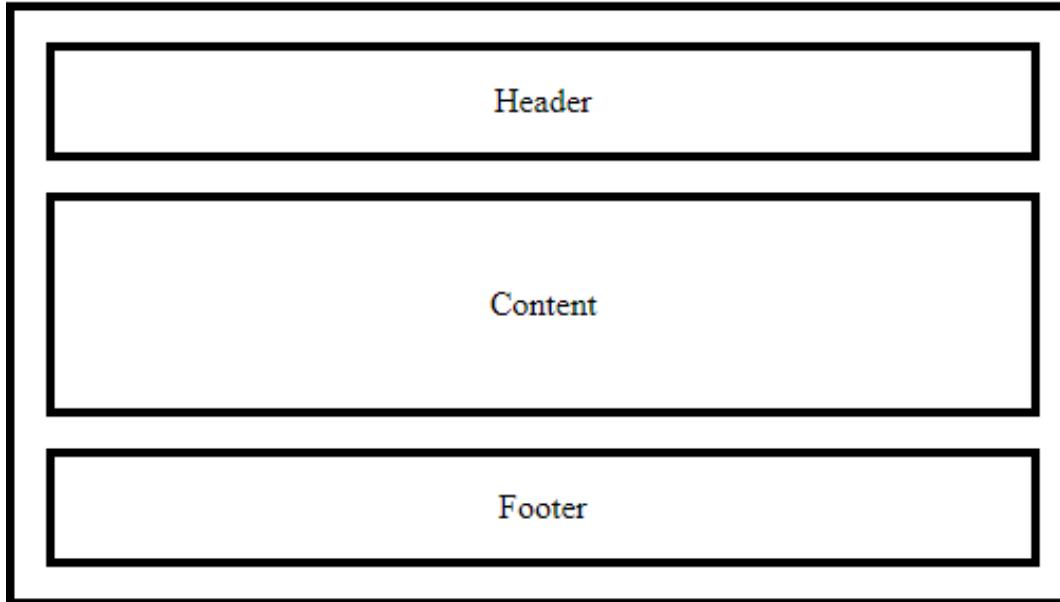
- Tiến hành dựng layout cho trang của bạn theo cấu trúc có dạng như sau, lưu ý bạn có thể đặt cấu trúc khác, nhưng nên theo tư duy chia thành các phần header, main, footer

```
<div class="container-fluid">
  <div class="header">
    HEADER
  </div>
  <div class="main">
    MAIN
  </div>
  <div class="footer">
    FOOTER
  </div>
</div>
```



# Các bước thực hiện

- Các thành phần của trang về mặt cơ bản sau khi thiết kế sẽ có 3 thành phần chính được hiển thị như sau





# Các bước thực hiện

- Sau khi có được layout, hoàn thiện trang sử dụng các kỹ năng HTML và CSS của bạn, theo tư duy sau:
  - Dựng hoàn chỉnh HTML trước rồi mới bắt đầu CSS
  - Trong quá trình dựng HTML, để ý đặt các class cho các thành phần cùng chung thuộc tính



# Thực hành



- Thực hành cắt giao diện sau, link file .psd bắt kì, ví dụ:

The screenshot shows a dark-themed website template titled "Bloom". At the top, there's a navigation bar with links: Home, About Us, Services, Portfolio, and Contact Us. Below the header, there's a section with placeholder text "Lorem ipsum dolor sit amet" followed by a paragraph of Latin text. To the right of this text are three circular maps with geographical data. A "Learn More" button is located below the text. The overall design is clean and modern.

This screenshot displays a section of a website with a heading "This is a Heading". Below the heading are three dark rectangular boxes, each labeled "Aliquam et iaculis sapien" and accompanied by a "Learn More" button. Each box also contains a short paragraph of placeholder text.

The screenshot shows a sidebar section. It includes a "Latest Posts" section with three items, each with a date (July 27), a thumbnail, and the title "Aliquam et iaculis sapien". Below this is an "About" section with some text and a "Learn More" button. On the far right, there's a "Stay Connected" section with links to Facebook, Twitter, RSS, and Google+.

# CSE391 - Nền tảng phát triển Web

## Bài 5: Bootstrap

dungkt@tlu.edu.vn

# Nội dung



- 1) Căn bản về Bootstrap
- 2) Fontawesome
- 3) Bootstrap Studio

# 1. Căn bản về Bootstrap





# Responsive là gì?

- Thể hiện cho tính tương thích của 1 website trên nhiều thiết bị
- Một cách đơn giản, 1 trang web không xuất hiện thanh cuộn ngang trên tất cả thiết bị thì web đó gọi là responsive
- Sử dụng thuộc tính css @media theo độ rộng thiết bị
- Thêm thẻ meta với nội dung sau vào thẻ head:
  - `<meta name="viewport" content="width=device-width, initial-scale=1">`



# Bootstrap là gì?



- Là 1 framework tích hợp sẵn cả HTML, CSS và Javascript
- Thiết kế theo hướng responsive nên tương thích với nhiều thiết bị
- Được tích hợp vào khá nhiều framework của PHP như Laravel, Zend, Cake



# Cài đặt Bootstrap



- Cách 1: Sử dụng link CDN (khuyên dùng khi triển khai thực tế)
- Cách 2: Download file bootstrap.min.css, jquery.min.js, bootstrap.min.js về local từ <https://getbootstrap.com/>

```
bootstrap/
  css/
    bootstrap.css
    bootstrap.css.map
    bootstrap.min.css
    bootstrap.min.css.map
    bootstrap-theme.css
    bootstrap-theme.css.map
    bootstrap-theme.min.css
    bootstrap-theme.min.css.map
  js/
    bootstrap.js
    bootstrap.min.js
    npm.js
  fonts/
    glyphicons-halflings-regular.eot
    glyphicons-halflings-regular.svg
    glyphicons-halflings-regular.ttf
    glyphicons-halflings-regular.woff
    glyphicons-halflings-regular.woff2
```

# Bootstrap Grid



	<b>Extra small</b> ≤576px	<b>Small</b> ≥576px	<b>Medium</b> ≥768px	<b>Large</b> ≥992px	<b>Extra large</b> ≥1200px
<b>Max container width</b>	None (auto)	540px	720px	960px	1140px
<b>Class prefix</b>	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
<b># of columns</b>	12				
<b>Gutter width</b>	30px (15px on each side of a column)				
<b>Nestable</b>	Yes				
<b>Column ordering</b>	Yes				

# HTML layout với Bootstrap



```
<div class="container">  
    <div class="row">  
        <div class="col-md-8 col-12">  
            HTML  
        </div>  
        <div class="col-md-4 col-12">  
            HTML  
        </div>  
    </div>  
</div>
```



# Bootstrap Components

- <https://getbootstrap.com/docs/4.1/components>

Primary Secondary Success Danger Warning Info Light Dark Link

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Copy

# Thực hành



A screenshot of a website with a dark gray background. At the top is a navigation bar with tabs: 'Project name' (highlighted in blue), 'Home', 'About', 'Contact', and 'Dropdown ▾'. Below the navigation is a large, bold headline 'Example headline.' followed by a short paragraph of text: 'Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.' A blue button labeled 'Sign up today' is centered below the text. At the bottom of the page are three small circular icons with the text '● ○ ○' underneath them.



## Heading

Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod. Nullam id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Praesent commodo cursus magna.

[View details ▾](#)



## Heading

Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh.

[View details ▾](#)



## Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details ▾](#)

## 2. Font Awesome





# Font Awesome là gì?

- Thư viện cung cấp sẵn các bộ icon phong phú cho website.
- Link minh họa các icon của bản free
  - <https://fontawesome.com/icons?d=gallery&m=free>
- Phiên bản hiện tại: 5.13.0
  - 1/5/2020



# Cài đặt Font Awesome



- **Cách 1:** Nhúng link css từ server của FontAwesome
  - <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/all.css" integrity="sha384-oS3vJWv+OUjzBfQzYUhtDYW+Pj2yciDJxpsK10YPAYjqT085Qq/1cq5FLXAZQ7Ay" crossorigin="anonymous">
- **Cách 2:** Download bản free, giải nén, import file all.min.css và toàn bộ thư mục webfonts vào project
  - <https://fontawesome.com/how-to-use/on-the-web/setup/hosting-font-awesome-yourself>

```
<head>
    <link href="/your-path-to-fontawesome/css/all.css" rel="stylesheet"> <!--load all styles -->
</head>
<body>
    <i class="fas fa-user"></i> <!-- uses solid style -->
    <i class="far fa-user"></i> <!-- uses regular style -->
    <i class="fal fa-user"></i> <!-- uses light style -->
    <!--brand icon-->
    <i class="fab fa-github-square"></i> <!-- uses brands style -->
</body>
```

# Sử dụng Font Awesome



- Link tham khảo
  - <https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use>
- Ví dụ:
  - `<i class="fa fa-angle-double-down"></i>`
- Ảnh hiển thị:

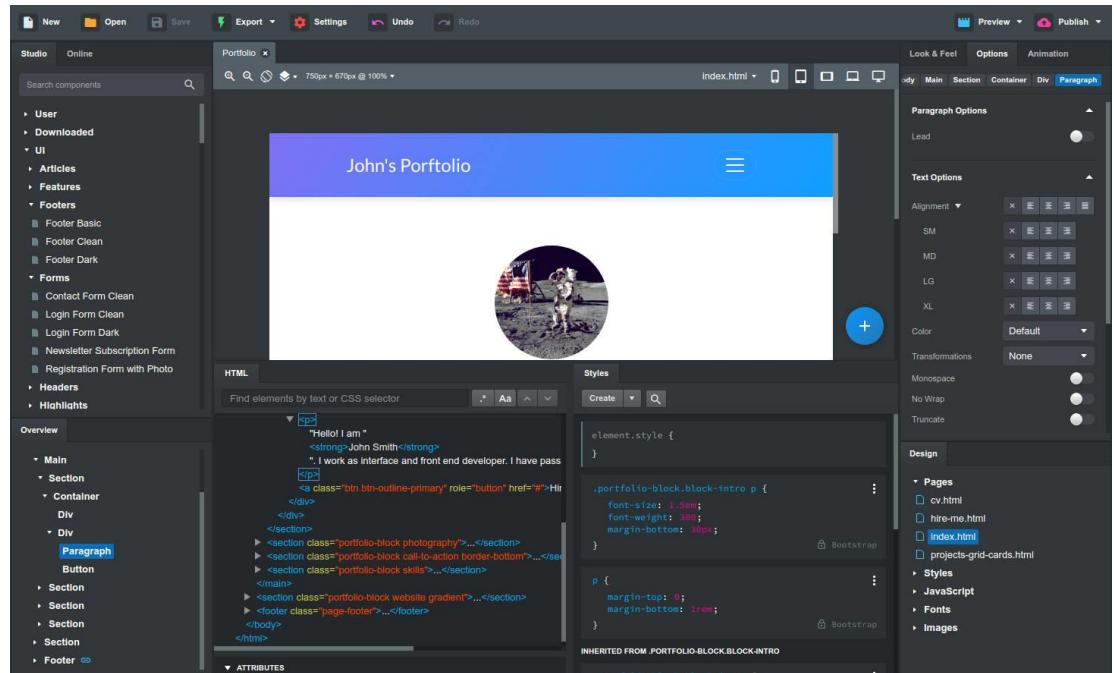


### 3. Bootstrap Studio



# Bootstrap Studio là gì?

- Công cụ hỗ trợ người dùng sử dụng thư viện Bootstrap dưới dạng kéo thả các thành phần và thiết kế bối cục trực quan.



# CSE391 - Nền tảng phát triển Web

## Bài 6: Javascript

dungkt@tlu.edu.vn

# 1. Javascript 01





# Nội dung

- 1) Khái niệm
- 2) Khai báo
- 3) Chương trình đầu tiên
- 4) Biến
- 5) Hằng
- 6) Kiểu dữ liệu biến
- 7) Hàm
- 8) Sự kiện

# Khái niệm



- Là ngôn ngữ lập trình frontend, giúp tăng tính tương tác của client với website
- Javascript có thể tương tác với HTML, tương tác với client thông qua các sự kiện như click chuột, gõ phím ...
- Javascript phân biệt hoa thường



# Khai báo



- Inline trong thẻ HTML (qua các sự kiện)

```
<a href="#" onclick="alert('abc')">Click vào đây để show popup</a>
```

- Nhúng trong file HTML

```
<script type="text/javascript">  
    // code javascript  
</script>
```

- Link file js bên ngoài

```
<script type="text/javascript" src="js/main.js"></script>
```

# Chương trình đầu tiên

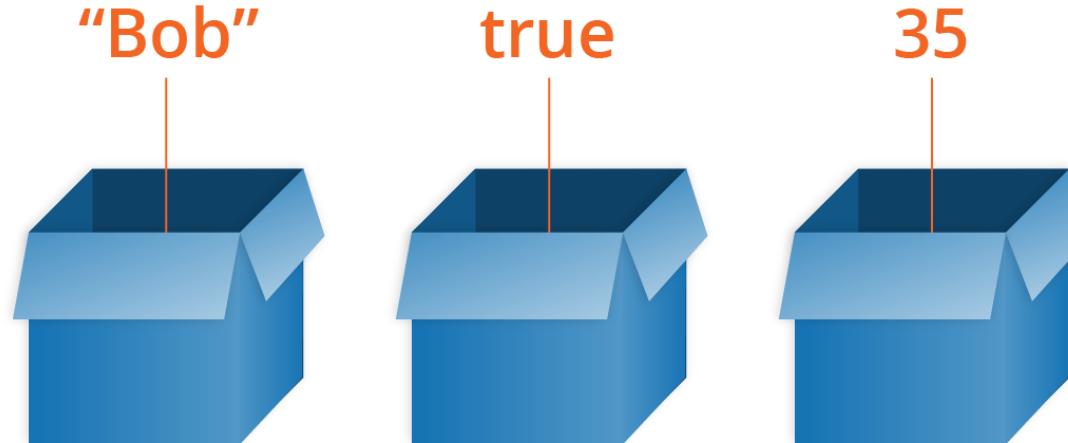


```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Javascript</title>
    </head>
    <body>
        <script type="text/javascript">
            document.write("Hello, World");
        </script>
    </body>
</html>
```



# Khái niệm Biến

- Về mặt vật lý là 1 tham chiếu đến bộ nhớ trong thiết bị
- Biến chứa các giá trị có thể thay đổi được



# Khai báo biến



- Sử dụng từ khóa var
  - var name;
  - var name, age;
  - var name = 'Manh';
- Quy tắc đặt tên
  - Tên biến phải bắt đầu bằng chữ hoặc kí tự gạch dưới \_.
  - Tên biến không được bắt đầu bằng số và không chứa các kí tự đặc biệt như &, \*, (, ).
  - Tên biến không được trùng với từ khóa như var, for, if...



# Khai báo biến

- Khai báo các biến Javascript theo mô tả sau:
  - username, giá trị nhanh
  - password, giá trị 123456
  - undefined, giá trị null
- Dự đoán xem các tên biến sau đang khai báo đúng hay sai:
  - \$abc, 5name, \_name, var, &variable
- Khai báo lần lượt 3 biến để thể hiện thông tin lần lượt về tên, tuổi, địa chỉ của bạn. Sau đó hiển thị ra màn hình sử dụng hàm document.write(<tên-biến>) có dạng sau, sử dụng toán tử + để nối chuỗi với biến mà bạn vừa khai báo:
  - Tên của bạn là: <tên> / Tuổi của bạn : <tuổi> / Địa chỉ của bạn : <địa-chỉ>

# Khai báo Hằng



- Cũng là biến tuy nhiên không thể thay đổi giá trị một khi nó được khai báo
- Sử dụng từ khóa const thay cho var
- Thường được đặt tên in hoa
- VD: const AGE = 12;

# Các kiểu dữ liệu



- 6 kiểu dữ liệu
  - Number //123, 1.23
  - String // 'abc', "123"
  - Boolean //true, false
  - Undefined //undefined
  - Array // [1, 2, 3, 'a']
  - Object // {name: 'Anh', age: 29}
- Có thể sử dụng hàm typeof <variable> để kiểm tra kiểu dữ liệu của biến
- Các kiểu dữ liệu nguyên thủy là number, string, boolean
- Chú ý: kiểu dữ liệu array trong Javascript về bản chất vẫn là các object

# Hàm



- Là 1 tập các dòng lệnh dùng để xử lý tác vụ nào đó
- Là khái niệm rất quan trọng trong lập trình
- Sử dụng hàm cho phép chia nhỏ chương trình thành các chức năng nhỏ hơn
- Tính chất quan trọng nhất của hàm là tính tái sử dụng



# Phân loại hàm

- Hàm xây dựng sẵn

VD: alert(), prompt()

- Hàm tự định nghĩa

```
function function_name(parameter_list){  
    //code Javascript  
    return value;  
}
```

- function\_name: tên hàm do bạn tự định nghĩa
- parameter\_list: danh sách các tham số truyền vào hàm nếu có
- return value: câu lệnh return này là tùy chọn, để thể hiện hàm này có mang giá trị nào không

# Gọi hàm



- Khi định nghĩa ra 1 hàm, thì hàm sẽ chưa được sử dụng cho đến khi nó được gọi
- Cú pháp

```
function_name(parameter_list);
```

- Ví dụ

```
function sum(a, b){  
    return a + b;  
}  
  
document.write("Tổng của 2 vs 5 = " + sum(2, 5));
```



# Các hàm thông dụng

- `document.write('<Noi-dung>');` //hiển thị text Noi-dung lên màn hình
- `console.log('<Nội-dung>');` //hiển thị text Nội-dung tại tab Console khi Inspect trình duyệt, dùng để debug
- `alert('<Nội-dung>');` //hiển thị text Nội-dung dưới dạng popup của trình duyệt
- `prompt('<Nội-dung>');` //hiển thị text Nội-dung dưới dạng popup nhập liệu
- `confirm('<Nội-dung>');` //hiển thị text Nội-dung dưới dạng popup xác nhận có thực thi hành động hay không
- `document.getElementById("<id-phần-tử-HTML>");` //lấy đối tượng có id là id-phần-tử-HTML
- `document.getElementById("<id-phần-tử-HTML>").innerHTML = '<Nội-dung-hiển-thị>';` //set nội dung Nội-dung-hiển-thị cho phần tử có id là id-phần-tử-HTML

# Sự kiện trong Javascript



- Là 1 hành động nào đó tác động lên đối tượng HTML mà ta có thể bắt được các hành động này thông qua các từ khóa mà javascript cung cấp
- Sự kiện trong JS có thể là click vào 1 nút nào đó, hoặc hover chuột qua 1 đoạn văn nào đó (thẻ p)....
- Để quyết định một sự kiện được kích hoạt hay không thì ta chỉ cần return false khi gọi sự kiện đó.
- Cú pháp khai báo sự kiện
  - Sử dụng inline trong thẻ HTML
    - <p onclick="<code-javascript>"></p>
  - Sử dụng phương thức eventListener
    - document.getElementById("myBtn").addEventListener("click", function);

# Các sự kiện thông dụng trong Javascript



Event Name	Description
onclick	Xảy ra khi click vào thẻ HTML
onchange	Xảy ra khi giá trị (value) của thẻ HTML bị thay đổi,
onmouseover	Xảy ra khi con trỏ chuột bắt đầu đi vào thẻ HTML
onmouseout	Xảy ra khi con trỏ chuột bắt đầu rời khỏi thẻ HTML
onmouseenter	Tương tự như onmouseover
onmouseleave	Tương tự như onmouseout

# Các sự kiện thông dụng trong Javascript



Event Name	Description
onmousemove	Xảy ra khi con chuột di chuyển bên trong thẻ HTML
onload	Xảy ra khi thẻ HTML bắt đầu chạy, thường sử dụng trong thẻ body, tương đương với jquery document ready
onkeyup	Xảy ra khi bạn gõ phím nhưng lúc bạn nhả phím ra sẽ được kích hoạt
onkeydown	Xảy ra khi gõ một phím bất kì vào ô input
onkeypress	Xảy ra khi bạn nhấn một phím vào ô input
onblur	Xảy ra khi con trỏ chuột rời khỏi ô input

## 2. Javascript 02



# Nội dung

- Toán tử
- Câu lệnh điều kiện
  - If
  - If...else
  - If...elseif...else
- Biểu thức switch...case
- Vòng lặp
  - For
  - While
  - Do...while
  - Từ khóa break - continue

# Toán tử số học



Toán tử	Mô tả	Ví dụ
+	Phép cộng	$A = 5 + 8$
-	Phép trừ	$A = 8 - 5$
*	Phép nhân	$A = 8 * 5$
/	Phép chia	$A = 20 / 5$
%	Phép chia lấy số dư	$10 \% 3 = 1$
++	Tăng lên một đơn vị	$++x$ sẽ trả về giá trị của x sau khi tăng. $x++$ sẽ trả về giá trị của x trước khi tăng.
--	Giảm một đơn vị	$--x$ sẽ trả về giá trị của x sau khi giảm. $x--$ sẽ trả về giá trị của x trước khi giảm..



# Toán tử so sánh

Toán tử	Mô tả	Ví dụ
<code>==</code>	Bằng. Trả về giá trị true nếu các toán hạng bằng nhau.	<code>a == b</code>
<code>!=</code>	Không bằng. Trả về giá trị true nếu các toán hạng không bằng nhau.	<code>a != 5</code>
<code>&gt;</code>	Lớn hơn. Trả về giá trị true nếu toán hạng trái lớn hơn toán hạng phải.	<code>a &gt; b</code>
<code>&gt;=</code>	Lớn hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái lớn hơn hoặc bằng toán hạng phải.	<code>a &gt;= b &gt;= c</code>
<code>&lt;</code>	Nhỏ hơn. Trả về giá trị true nếu toán hạng trái nhỏ hơn toán hạng phải.	<code>a &lt; b</code>
<code>&lt;=</code>	Nhỏ hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái nhỏ hơn hoặc bằng toán hạng phải.	<code>a &lt;= b &lt;= c</code>

# Toán tử logic



Toán tử	Giá trị	Mô tả
And ( && )	expr1 && expr2	Trả về true khi cả 2 biểu thức expr1 và expr2 trả về true
Or (    )	expr1    expr2	Trả về true khi có ít nhất 1 trong 2 biểu thức expr1 và expr2 trả về true
Not ( ! )	!expr	Trả về giá trị false nếu biểu thức đúng và trả về giá trị true nếu biểu thức sai.

# Toán tử gán



Toán tử	Ví dụ	Viết đầy đủ
=	$x = y;$	
+=	$x += y;$	$x = x + y;$
-=	$x -= y;$	$x = x - y;$
*=	$x *= y;$	$x = x * y;$
/=	$x /= y;$	$x = x / y;$
%=	$x %= y;$	$x = x \% y;$

# Câu trúc điều kiện IF



- Cú pháp

```
if (expression){  
    Statement(s) to be executed if expression is true  
}
```

- Chỉ kiểm tra duy nhất 1 trường hợp khi biểu thức điều kiện expression TRUE



# Câu trúc điều kiện IF

- Cú pháp

```
if (expression){  
    Statement(s) to be executed if expression is true  
}  
else {  
    Statement(s) to be executed if expression is false  
}
```

- Kiểm tra 2 trường hợp dựa vào tính đúng sai của biểu thức điều kiện expression



# Câu trúc điều kiện IF

- Cú pháp

```
if (expression 1){  
    Statement(s) to be executed if expression 1 is true  
}  
else if (expression 2){  
    Statement(s) to be executed if expression 2 is true  
}  
else if (expression 3){  
    Statement(s) to be executed if expression 3 is true  
}  
else {  
    Statement(s) to be executed if no expression is true  
}
```

- Kiểm tra > 2 trường hợp dựa vào tính đúng sai của biểu thức điều kiện expression tương ứng

# Biểu thức switch .. case



- Cú pháp

```
switch(expression){  
    case condition 1: statement(s)  
    break;  
    case condition 2: statement(s)  
    break;  
    ...  
    case condition n: statement(s)  
    break;  
    default: statement(s)  
}
```

Có thể thay thế khi sử dụng quá nhiều if..elseif..else

Expression sử dụng được các kiểu dữ liệu nguyên thủy (number, string, boolean)

# Vòng lặp FOR



- Cú pháp

```
for (initialization; test condition; iteration statement){  
    Statement(s) to be executed if test condition is true  
}
```

- Ví dụ:

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

- Thường sử dụng khi biết trước số lần lặp
- Cần chú ý về điều kiện dừng vòng lặp, tránh vòng lặp vô hạn

# Vòng lặp WHILE



- Cú pháp

```
while (expression){  
    Statement(s) to be executed if expression is true  
}
```

- Ví dụ: `while(i > 10) {}`
- Thường sử dụng khi không biết trước số lần lặp
- Cần chú ý về điều kiện dừng vòng lặp, tránh vòng lặp vô hạn



# Vòng lặp DO .. WHILE

- Cú pháp

```
do {  
    Statement(s) to be executed;  
} while (expression);
```

- Ví dụ: do {

```
    document.write('abc');  
}  
while(i > 10)
```

- Bản chất giống vòng lặp while
- Khác biệt duy nhất là vòng lặp luôn thực hiện logic code ít nhất 1 lần, cho dù điều kiện while là FALSE
- Cần chú ý về điều kiện dừng vòng lặp, tránh vòng lặp vô hạn



# Từ khóa Break, Continue

- Được sử dụng chủ yếu trong các vòng lặp
- **break:**
  - Kết thúc vòng lặp mà không quan tâm đến điều kiện của vòng lặp đang TRUE hay FALSE
- **continue:**
  - Nhảy tới lần lặp kế tiếp, đồng thời bỏ qua các dòng lệnh phía sau nó trong vòng lặp hiện tại

# CSE391 - Nền tảng phát triển Web

## Bài 7: Javascript - jQuery

dungkt@tlu.edu.vn



# Nội dung

- 1) Khái niệm
- 2) Cài đặt
- 3) Chương trình đầu tiên
- 4) DOM
- 5) Quy trình biên dịch Javascript



# Nội dung

## 6) Các module trong jQuery

- jQuery Selector
- jQuery Attribute
- jQuery DOM
- jQuery Event
- jQuery Effect
- jQuery Ajax

## 7) JQuery với Bootstrap



# Khái niệm

- Là thư viện được xây dựng từ Javascript
- jQuery - “viết ít, làm nhiều”
- Là thư viện được tích hợp nhiều nhất trong các website hiện nay
- Thư viện Bootstrap cũng đã được tích hợp sẵn jQuery

A screenshot of a code editor window titled "JS". The code is written in jQuery. It includes CSS-like styling for an ul element and a click event handler for an ul element. To the right of the code editor is the official jQuery logo, which consists of a stylized orange crescent moon above the word "jQuery".

```
$('ul').children().slice(1,-2).css('background-color','salmon');
$('ul').children().first().has('b').css('background-color','violet');
$('ul').children().last().has('em').css('background-color','lime');

$('ul').click(function( event ) {
    var target = $( event.target );
    if ( target.is( "b" ) ) {
        target.css( "background-color", "red" );
    }
});
```

# Cài đặt



- Tải về tệp jquery.min.js từ <https://jquery.com/> và sử dụng offline trong Project.
- Sử dụng link bên ngoài như:
  - Google CDN
    - <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  - Microsoft CDN
  - CDNJS CDN
  - jsDelivr CDN

# Chương trình đầu tiên



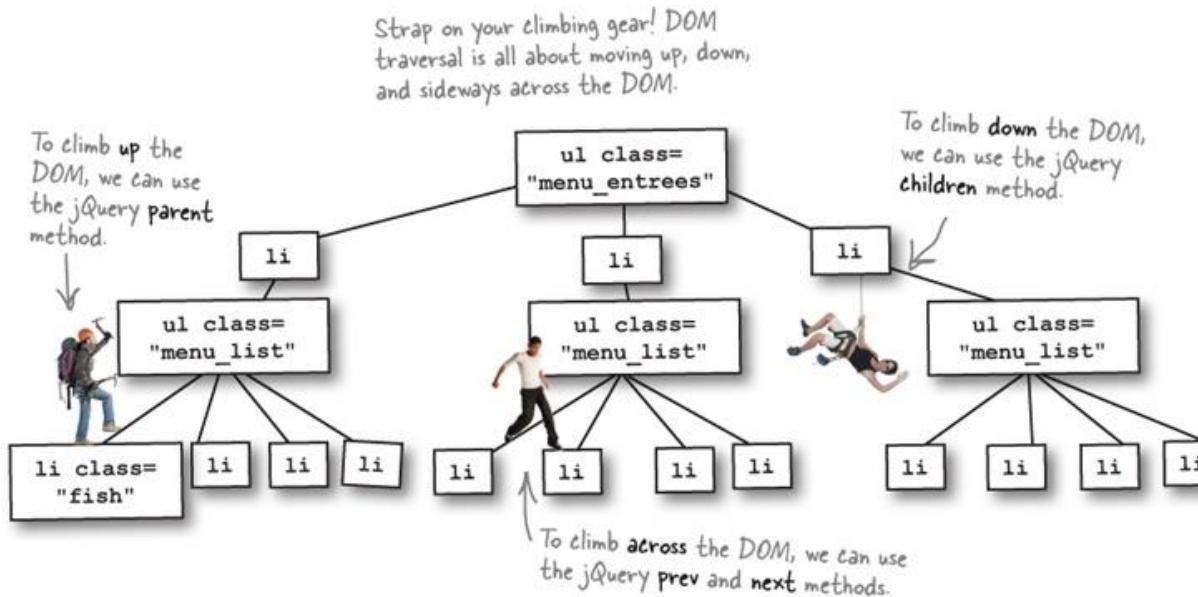
- jQuery luôn bắt đầu bằng từ khóa **\$** hoặc **jQuery**
- Cơ chế hàm callback - hàm đóng vai trò như 1 tham số - thường xuyên được sử dụng trong lập trình

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <script src="js/jquery-3.4.1.js"></script>
        <title>Title</title>
    </head>
    <body>
        <script type="text/javascript">
            $(document).ready(function () {
                document.write("Hello, World");
            })
        </script>
    </body>
</html>
```

# DOM



- Document Object Model
- Các thẻ HTML được quản lý bởi DOM



# Quy trình biên dịch



- File HTML được biên dịch từ trên xuống, từ trái qua phải
- Nguyên tắc khi code JS: chờ DOM load xong rồi mới thực thi các mã JS
- Sự kiện document.ready trong jQuery tương đương với window.onload trong Javascript
- Luôn đặt code JS trong document.ready (jQuery) hoặc window.onload (Javascript)



# Module jQuery – Bộ chọn

- Xác định phần tử HTML sẽ thao tác, cơ chế giống selector CSS
  - Selector by tag name
    - \$(p)
  - Selector by ID
    - \$('#some-id')
  - Selector by class name
    - \$('.some-class')
  - Selector multiple
    - \$('.some-class, #some-id')

# Module jQuery - Thuộc tính



- Thao tác với các thuộc tính của thẻ HTML như id, class, title, src, ... Các phương thức thường sử dụng
  - selector.attr(name): lấy giá trị của thuộc tính name
  - selector.attr(name, value): set giá trị value cho thuộc tính name
  - selector.removeAttr(name): xóa thuộc tính name
  - selector.hasClass(className): trả về true nếu có class className, ngược lại là false
  - selector.removeClass(className): xóa class className
  - selector.html(): lấy nội dung dạng html của selector
  - selector.html(value): set nội dung value cho selector
  - selector.text(): lấy nội dung dạng text của selector
  - selector.text(value): set nội dung text cho selector
  - selector.val(): lấy giá trị của thuộc tính value trong input form
  - selector.val(value): set giá trị value cho thuộc tính value trong input form

# Module jQuery - DOM



- Thao tác với các thành phần DOM
- Các phương thức thường sử dụng
  - selector.replaceWith(element): thay thế nội dung selector bằng nội dung element
  - selector.remove(): xóa selector
  - selector.after(element): thêm phần tử element ở vị trí phía sau và ngang hàng với selector
  - selector.before(element): thêm phần tử element ở vị trí trước và ngang hàng với selector
  - selector.append(element): thêm phần tử element là con của selector, có vị trí đầu tiên
  - selector.prepend(element): thêm phần tử element là con của selector, có vị trí cuối cùng

# Module jQuery - Sự kiện (Event)



- Thao tác với các sự kiện
- jQuery viết lại cách gọi các sự kiện, nhưng về bản chất vẫn hoạt động sử dụng cơ chế của Javascript
- Có thể có 2 cách viết

```
$('#paragraph').click(function () {  
    console.log("clicked");  
});  
  
$('#paragraph').bind('click', function () {  
    console.log("clicked");  
})
```

# Module jQuery - Hiệu ứng (Effect)



- Cung cấp các hàm liên quan đến hiệu ứng cho phần tử
- Một số hàm thông dụng:
  - selector.show(speed): hiển thị selector với tốc độ speed tính bằng milliseconds
  - selector.hide(speed): ẩn selector với tốc độ speed tính bằng milliseconds
  - selector.toggle(speed): hiển thị/ẩn selector dựa vào trạng thái hiện tại, nếu đang hiển thị -> ẩn và ngược lại
  - selector.animate(properties, speed): tạo hiệu ứng chuyển động cho selector dựa vào các properties với tốc độ speed tính bằng milliseconds
  - selector.fadeIn(speed): hiển thị selector theo cơ chế fade với tốc độ speed
  - selector.fadeOut(speed): ẩn selector theo cơ chế fade với tốc độ speed

# Module jQuery - Ajax



- Ajax - Asynchronous JavaScript and XML, là kỹ thuật cho phép gửi nhận dữ liệu tới server mà không cần load lại trang
- Thường được sử dụng trong PHP, theo mô hình client - server
- Các phương thức thường gặp
  - selector.load(url, [data]): load dữ liệu từ url, có thể gửi dữ liệu data hoặc không
  - \$.ajax(options): load data theo cơ chế ajax
  - \$.get(options): load data theo cơ chế ajax sử dụng phương thức GET
  - \$.post(options): sử dụng phương thức POST



# jQuery - Bootstrap

## CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-"> Copy
```

## JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use [jQuery's slim build](#), but the full version is also supported.

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkVYIK3UENzmM7KCkRr/rE9/Qpg6a"> Copy
```

# CSE391 - Nền tảng phát triển Web

## Bài 8: Javascript - jQuery nâng cao

dungkt@tlu.edu.vn

# Nội dung



- 1) Biểu thức chính qui
- 2) jQuery Validate plugin
- 3) jQuery UI

# 1. Biểu thức chính qui





# Nội dung

- 1) Khái niệm
- 2) Khởi tạo
- 3) Phương thức test
- 4) Một số điều kiện sẵn có
- 5) Các điều kiện hay sử dụng
- 6) Setting Regex
- 7) Thực hành

# Khái niệm



- Hiểu đơn giản RegEx là 1 pattern (mẫu) cho phép bạn kiểm tra 1 chuỗi ký tự như email, mật khẩu .v.v có khớp với pattern mà bạn quy định từ trước hay không
- Regex rất khó học, khó viết, khó nhớ, khó sửa. tuy nhiên có những việc mà chỉ sử dụng regex thì mới có thể thực hiện được
- VD: kiểm tra xem 1 chuỗi bất kỳ có kết thúc bằng .com, .org hay không chẳng hạn
  - abc.com => true
  - abc.org => true
  - abc.comvn => false
  - => chuỗi regex này sẽ có dạng `/(\.com|\.\org)/`

# Ví dụ



/ (\.com | \.org) \$ /i

“or”  
↓  
/ ( \.com | \.org ) \$ /i  
          |  
          | “Escape the special character “dot””  
          | “Appear at the end”  
          | “Appear once”  
          ↑

- Ký tự \ đặt trước các ký tự đặc biệt để hiện thị chính xác ký tự đó
- Ký tự | là điều kiện OR các biểu thức Regex
- Ký tự \$ là điều kiện phải xuất hiện ở cuối chuỗi, ngược lại sẽ có ký tự ^ là phải xuất hiện ở đầu chuỗi
- Phần nằm giữa ký tự // là nơi viết các biểu thức Regex của bạn
- Ký tự i ở cuối là setting cho phép không phân biệt hoa thường khi sử dụng Regex



# Khởi tạo

- Trong Javascript, Regex là 1 object, được khởi tạo theo 2 cách sau
  - var regex = new RegExp(pattern);
  - var regex =/abc/; //thường sử dụng cách này

# Phương thức Test



- Kiểm tra xem chuỗi ban đầu có khớp với biểu thức regex hay không, trả về true nếu khớp, ngược lại trả về false
  - var regex = /abc/;
  - console.log(regex.test("abc abc 123")); //true
  - console.log(regex.test("bca")); //false

# Phương thức Exec



- Tìm kiếm trong chuỗi ban đầu mà khớp với regex. Nếu tìm thấy sẽ trả về mảng kết quả, ngược lại trả về null
  - var regex = /abc/;
  - console.log(regex.exec("abc abc 123"));



# Một số điều kiện có sẵn

Ký tự	Giải thích
\d	Tương đương với [0-9], chứa ký tự số
\D	Tương đương với [^0-9], chứa ký tự không phải số
\w	Tương đương [A-Za-z0-9], chứa ký tự là chữ hoặc số
\W	Tương đương [^A-Za-z0-9], chứa ký tự không phải chữ và số
\s	chứa ký tự khoảng trắng như space, tab(\t), newline (\n)
\S	Ngược lại với \s, không chứa ký tự khoảng trắng



# Các điều kiện hay sử dụng

- Phải xuất hiện ở đầu chuỗi: ^
  - var regex = /^nvanh/;
  - console.log(regex.test("nvanh is")); //true
  - console.log(regex.test("is nvanh is")); //false

# Các điều kiện hay sử dụng



- Phải xuất hiện ở cuối chuỗi: \$
  - var regex = /nvanh\$/;
  - console.log(regex.test("is nvanh")); //true
  - console.log(regex.test("nvanh is")); //false



# Các điều kiện hay sử dụng

- So sánh tuyệt đối: kết hợp 2 ký tự ^ và \$
  - var regex = /^nvanh\$/;
  - console.log(regex.test("nvanh")); //true
  - console.log(regex.test("nvanh is")); //false



# Các điều kiện hay sử dụng

- Bắt đầu bằng 1 pattern, kết thúc là 1 pattern: .\*
  - Ví dụ: bắt đầu bằng Name, kết thúc là Anh
  - var regex = /^Name.\*Anh\$/;
  - console.log(regex.test("Name is Anh")); //true
  - console.log(regex.test("nvanh is")); //false



# Các điều kiện hay sử dụng

- Nằm trong khoảng: [ Điểm\_bắt\_đầu-Điểm\_kết\_thúc ]
  - [a-z]: các ký tự từ a,b,c ... , x,y,z
  - [b-e]: các ký tự b,c,d,e
  - [A-Z]: các ký tự A,B,C ..., X,Y,Z
  - [0-9]: các ký tự 0,1,2 ... 9
  - [a-zA-Z0-9]: các ký tự a-z, A-Z, 0-9
- Ví dụ
  - var regex = /[a-z]/;
  - console.log(regex.test("Name is Anh 123")); //true
  - console.log(regex.test("1212")); //false



# Các điều kiện hay sử dụng

- Đảo ngược kết quả: [^]
  - var regex = /^[^0-9]/;
  - console.log(regex.test("anh is nvanh")); //true
  - console.log(regex.test("12")); //false



# Các điều kiện hay sử dụng

- Điều kiện hoặc: |
  - var regex = /anh|nvanh/;
  - console.log(regex.test("anh is nvanh")); //true
  - console.log(regex.test("an")); //false



# Các điều kiện hay sử dụng

- Xuất hiện đúng x lần: {x}
  - var regex = /^[0-9]{3}\$/;
  - console.log(regex.test("12")); //false
  - console.log(regex.test("123")); //true
  - console.log(regex.test("1234")); //false



# Các điều kiện hay sử dụng

- Xuất hiện từ x đến y lần: {x,y}
  - var regex = /^[0-9]{3,5}\$/;
  - console.log(regex.test("12")); //false
  - console.log(regex.test("123")); //true
  - console.log(regex.test("1234")); //true
- Nếu không set y, thì hiểu là >=x lần
  - var regex = /^[0-9]{3,}\$/;
  - console.log(regex.test("12")); //false
  - console.log(regex.test("123")); //true
  - console.log(regex.test("123412121")); //true



# Các điều kiện hay sử dụng

- Điều kiện không bắt buộc: ?
  - var regex = /^[0-9]{0,}?\$/;
  - console.log(regex.test("")); //true
  - console.log(regex.test("123")); //true
  - console.log(regex.test("123412121")); //true

# Các điều kiện hay sử dụng



- Nhóm điều kiện: ()
  - var regex = /^(\d{2})+\$/;
  - console.log(regex.test("1")); //false
  - console.log(regex.test("3412")); //true
  - console.log(regex.test("232")); //false



# Các điều kiện hay sử dụng

- Kiểm tra xem chuỗi có đáp ứng điều kiện ngay sau nó hay không: (?:=)
- Ví dụ: kiểm tra xem chuỗi có chứa ký tự ngay sau từ is là abc hoặc def không
  - var pattern = /is(?:=abc|def)/;
  - console.log(pattern.test('Hello isabc')); //true
  - console.log(pattern.test("Hello isdef")); //true
  - console.log(pattern.test("Hello is abc")); //false
- Ví dụ: Kiểm tra 1 chuỗi có độ dài từ 5 đến 10 ký tự, và phải chứa chuỗi abc hoặc def
  - var pattern = /^(?=.\*\w{3}def)[a-z0-9]{5,9}\\$/;
  - console.log(pattern.test('1212isabc'));
  - console.log(pattern.test('11isabc'));

# Thiết lập RegExp



- /abc/i

Ký tự	Giải thích
g	kiểm tra điều kiện nhiều lần
i	không phân biệt hoa thường
m	cho phép kiểm tra xuống dòng

## 2. jQuery validate plugin

# Nội dung



- Khái niệm
- Khai báo
- Cách sử dụng



# Khái niệm

- Là thư viện viết sẵn dựa trên jQuery
- Giúp validate dữ liệu theo kiểu realtime, nghĩa là thông báo lỗi ngay khi user nhập sai mà chưa cần submit

**Account name**

**This field is required.**

**Save**

# Cài đặt



- Có thể download về hoặc nhúng link online (CDN)
- Phải nhúng thư viện jQuery trước
- Nhúng sử dụng link CDN

```
<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
```

```
<script type="text/javascript"
```

```
src="http://ajax.aspnetcdn.com/ajax/jquery.validate/1.13.1/jquery.validate.min.js"></script>
```

# Sử dụng

```
<script type="text/javascript">
    $(document).ready(function () {
        var object = {
            errorPlacement: function(error, element) {
                error.insertAfter(element);
                error.wrap("<span class='l-error-text'>");
            },
            rules: {
                username: {
                    required: true,
                },
                age: {
                    required: true
                },
            },
            messages: {
                username: {
                    required: "vui lòng nhập tên"
                },
                age: {
                    required: "vui lòng nhập tuổi"
                }
            }
        });
        $('#form').validate(object);
    });
</script>
```



# Sử dụng - Các luật



required	Không được bỏ trống
remote	Gửi yêu cầu về Web Server để xác thực
minlength	Độ dài tối thiểu
min	Số tối thiểu
max	Số tối đa
range	Số tối thiểu từ x tới y
email	Xác thực định dạng Email
url	Xác thực định dạng URL
date	Xác thực định dạng ngày tháng
number	Phải là số, bao gồm số thập phân
equalTo	Phải trùng với phần tử nào đó

# Thực hành



- Sử dụng jquery validate để validate form như hình sau, nếu trường Account name để trống sẽ báo lỗi như trong hình

**Account name**

This field is required.

**Save**

## 2. jQuery UI



# Khái niệm



- jQuery UI - jQuery User Interface
- Là 1 thư viện miễn phí được xây dựng dựa trên jQuery, cung cấp những ứng dụng, widget, theme nâng cao hơn.
- Ví dụ về mặt hiệu ứng, trong khi jQuery chỉ có phương thức `$(selector).show(time, callback)` thì jQuery UI bổ sung 1 đống các extend vào như sau: `$(selector).show("EFFECT NAME", time, callback)`



# Sử dụng

- Truy cập trang chủ jQuery UI: <https://jqueryui.com/>
- Trải nghiệm các component, click view source để xem code

The screenshot shows the 'Datepicker' section of the jQuery UI demo page. On the left, there are two navigation menus: 'Interactions' (Draggable, Droppable, Resizable, Selectable, Sortable) and 'Widgets' (Accordion, Autocomplete, Button, Checkboxradio, Controlgroup, Datepicker, Dialog, Menu, Progressbar, Selectmenu, Slider, Spinner). The 'Datepicker' item in the 'Widgets' menu is highlighted with a red box. The main content area is titled 'Datepicker' and contains the text 'Select a date from a popup or inline calendar'. Below this is a form input field labeled 'Date:' with a red box around it. At the bottom of the content area, there is explanatory text about the datepicker's functionality and a 'view source' link, which is also highlighted with a red box.