

CSE391 – Nền tảng phát triển Web

Bài 2: HTML

dungkt@tlu.edu.vn

Nội dung

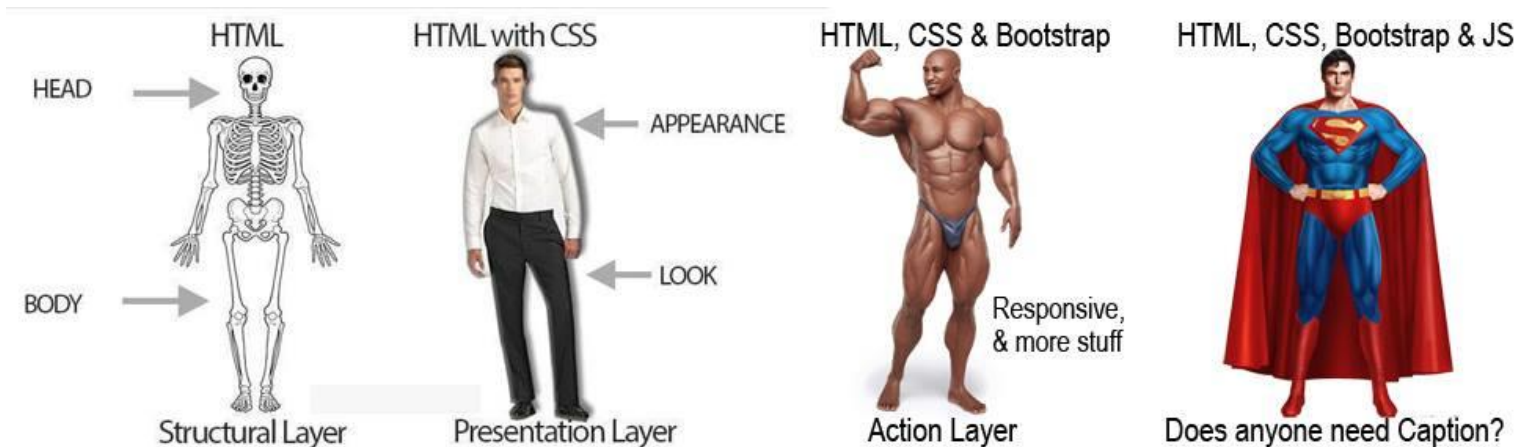


- 1) Khái niệm HTML
- 2) Cấu trúc trang HTML căn bản
- 3) Các thẻ HTML cơ bản
- 4) Các thẻ HTML5

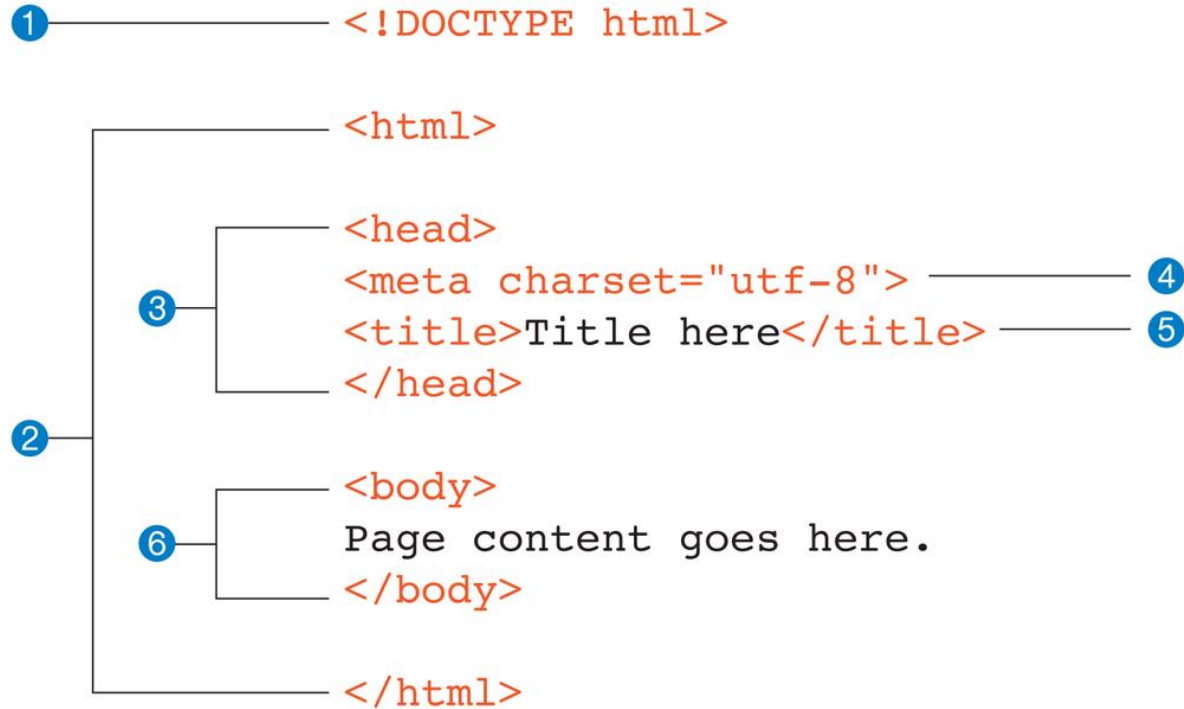
Khái niệm HTML



- 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
- `
` : 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



- `` : 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
- `` : 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



- `` : 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



- ``, ``, ``: Các thẻ liệt kê
 - `` : sắp xếp không có thứ tự, có thuộc tính type nhận các giá trị square, disc, circle
 - `` : sắp xếp có thứ tự
 - `` : tương ứng với 1 phần tử con trong `` và ``
- `<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 : dung 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

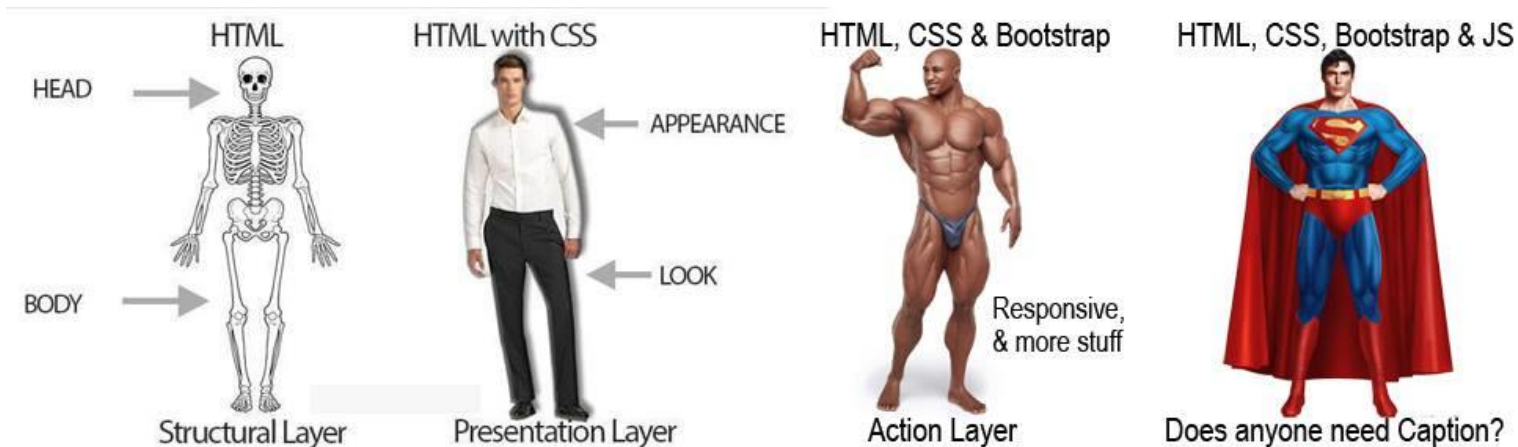


- 1) Khái niệm
- 2) Cú pháp khai báo
- 3) Selector
- 4) Nhúng file CSS
- 5) Thuộc tính CSS căn bản
- 6) Thành phần inline và block

Khái niệm CSS



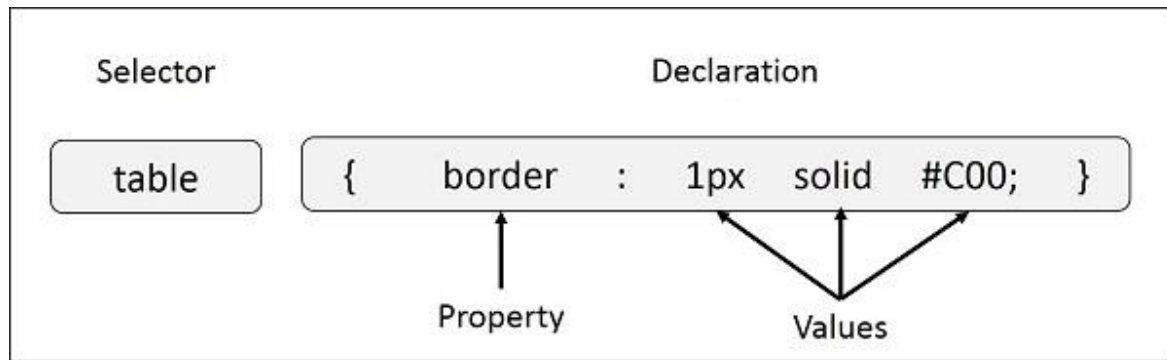
- CSS - Cascading Style Sheets
 - Ngôn ngữ hướng tới design Web bằng cách định dạng lại thẻ HTML



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

Nhúng 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: , <i>, <u>, <a>,
- Một số thẻ block: <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <div>

CSE391 – Nền tảng phát triển Web

Bài 4: PSD sang HTML và CSS

dungkt@tlu.edu.vn

Nội dung

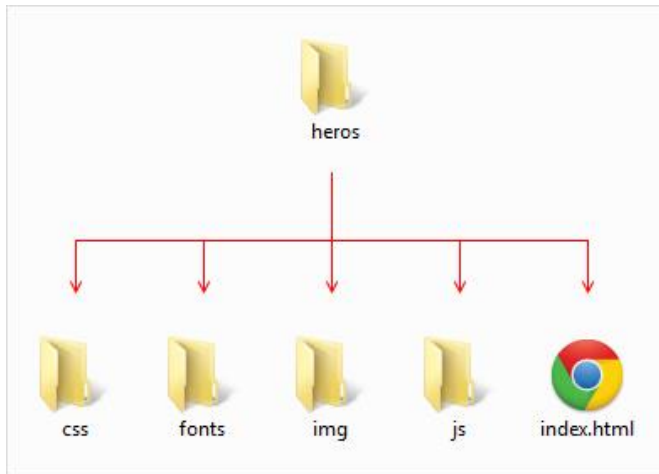


- 1) Các bước thực hiện
- 2) Thực hành

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.otf là font bạn vừa tải về

```
@font-face {  
    font-family: 'PlayfairDisplay-Black';  
    src: url("../fonts/PlayfairDisplay-Black.otf");  
}
```

```
/*set font cho phần tử HTML*/  
body{  
    font-family: 'PlayfairDisplay-Black', sans-serif;  
}
```

Các bước thực hiện



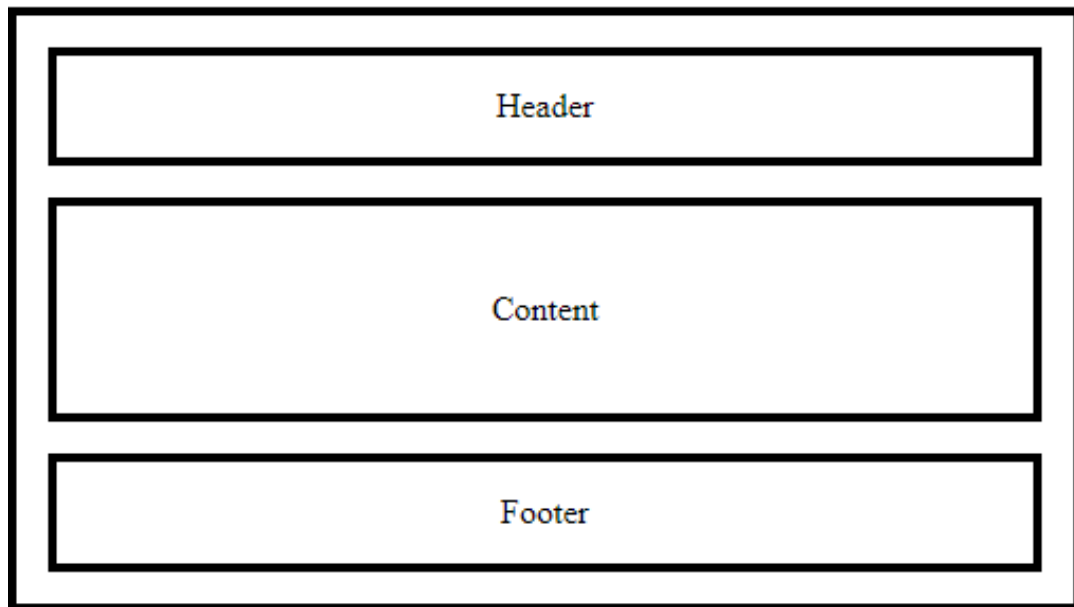
- Tiến hành dựng layout cho trang của bạn theo cấu trúc có dạng như sau, lưu ý bạn có thể đặt cấu trúc khác, nhưng nên theo tư duy chia thành các phần header, main, footer

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

Các bước thực hiện



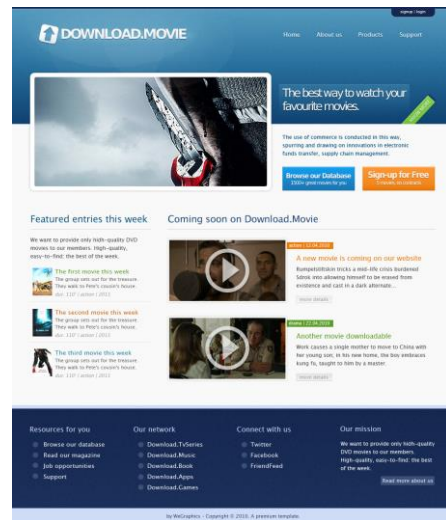
- Các thành phần của trang về mặt cơ bản sau khi thiết kế sẽ có 3 thành phần chính được hiển thị như sau



Các bước thực hiện



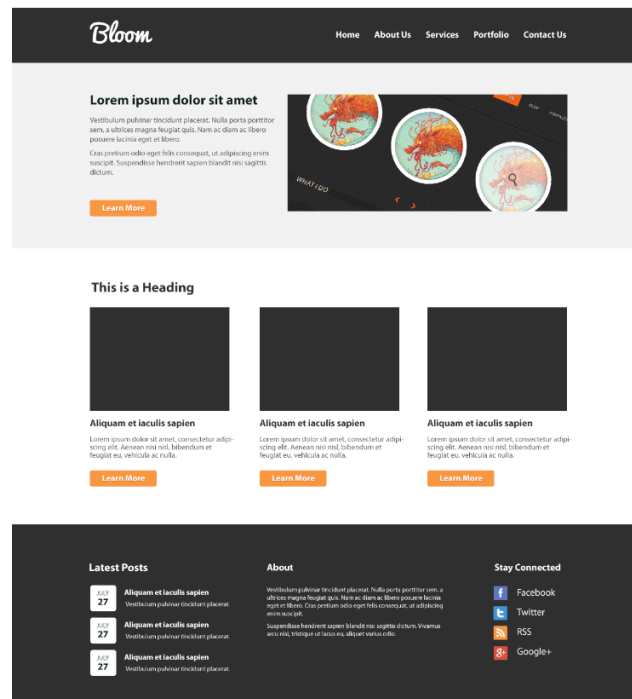
- Sau khi có được layout, hoàn thiện trang sử dụng các kỹ năng HTML và CSS của bạn, theo tư duy sau:
 - Dựng hoàn chỉnh HTML trước rồi mới bắt đầu CSS
 - Trong quá trình dựng HTML, để ý đặt các class cho các thành phần cùng chung thuộc tính



Thực hành



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



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 (khuyến 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/  
│   ├── glyphs-halflings-regular.eot  
│   ├── glyphs-halflings-regular.svg  
│   ├── glyphs-halflings-regular.ttf  
│   ├── glyphs-halflings-regular.woff  
│   └── glyphs-halflings-regular.woff2
```

Bootstrap Grid



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

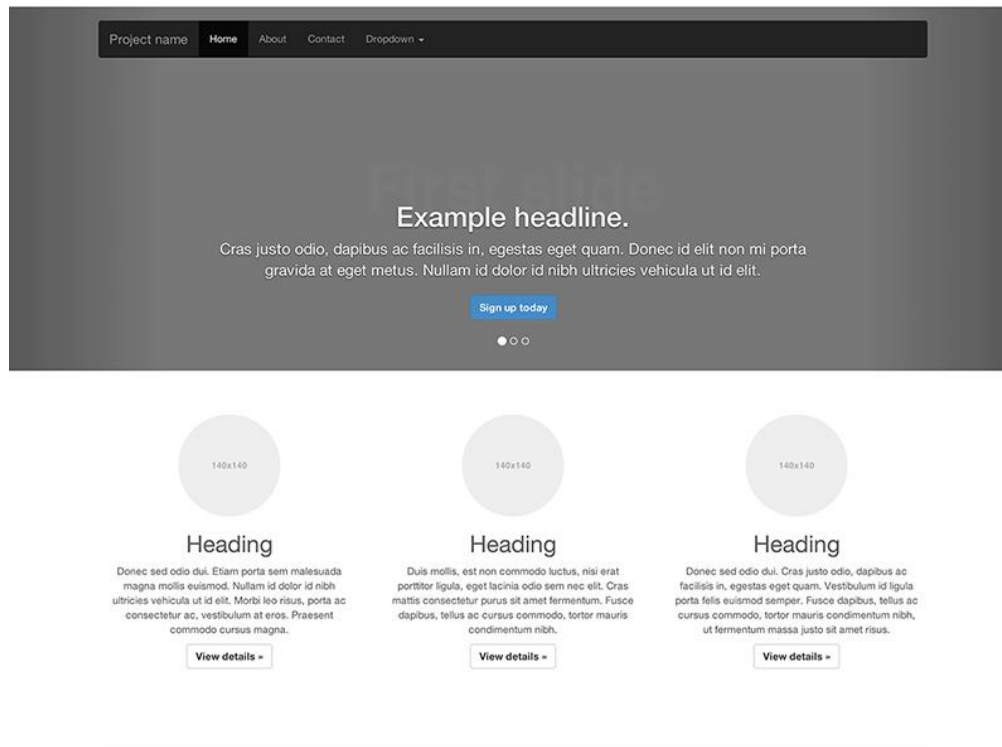


```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</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

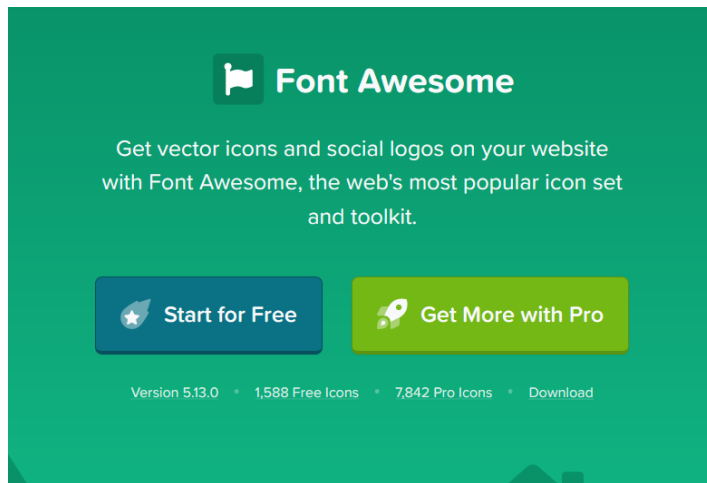


2. Font Awesome

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:

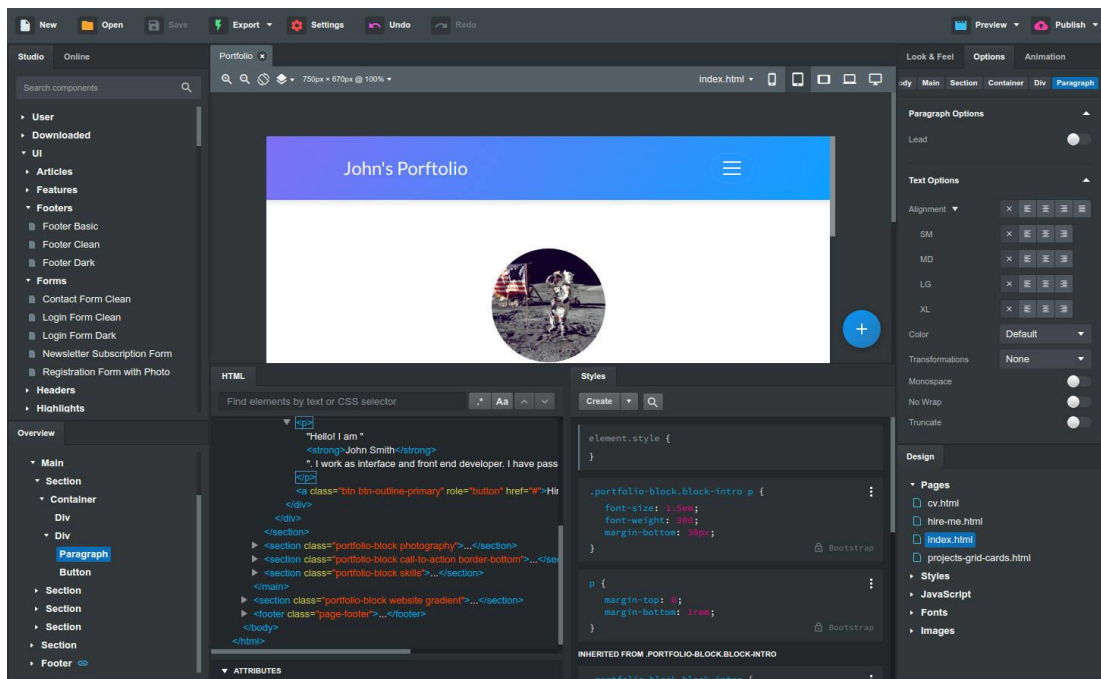


3. Bootstrap Studio

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 bố 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 (thông qua các sự kiện)
`Click vào đây để show popup`
- 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

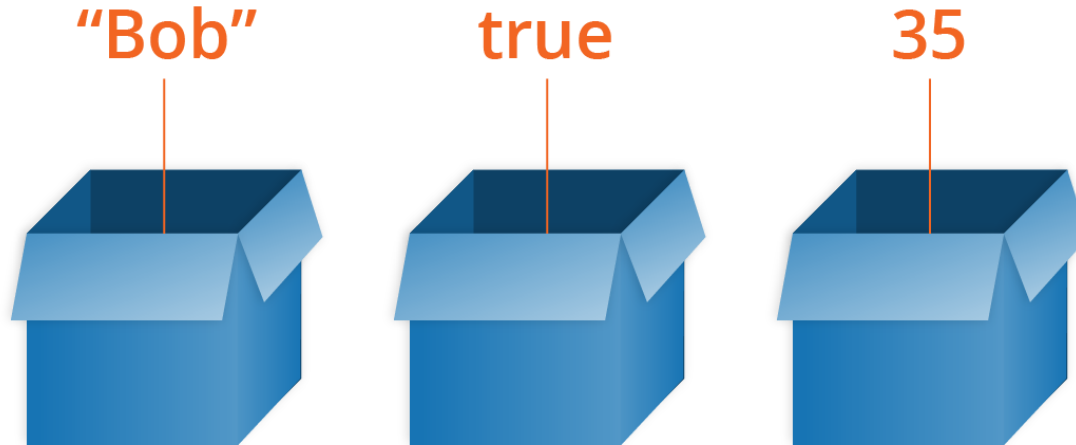


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Javascript</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write("Hello, World");
    </script>
  </body>
</html>
```

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 nvanh
 - 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('<Nôi-dung>');` //hiển thị text Noi-dung tại tab Console khi Inspect trình duyệt, dùng để debug
- `alert('<Nôi-dung>');` //hiển thị text Noi-dung dưới dạng popup của trình duyệt
- `prompt('<Nôi-dung>');` //hiển thị text Noi-dung dưới dạng popup nhập liệu
- `confirm('<Nôi-dung>');` //hiển thị text Noi-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 `addEventListener`
 - `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>></code>	Lớn hơn. Trả về giá trị true nếu toán hạng trái lớn hơn toán hạng phải.	<code>a > b</code>
<code>>=</code>	Lớn hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái lớn hơn hoặc bằng toán hạng phải.	<code>a >= b >= c</code>
<code><</code>	Nhỏ hơn. Trả về giá trị true nếu toán hạng trái nhỏ hơn toán hạng phải.	<code>a < b</code>
<code><=</code>	Nhỏ hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái nhỏ hơn hoặc bằng toán hạng phải.	<code>a <= b <= 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
=	<code>x = y;</code>	
+=	<code>x += y;</code>	<code>x = x + y;</code>
-=	<code>x -= y;</code>	<code>x = x - y;</code>
*=	<code>x *= y;</code>	<code>x = x * y;</code>
/=	<code>x /= y;</code>	<code>x = x / y;</code>
%=	<code>x %= y;</code>	<code>x = x % y;</code>

Cấu 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

Cấu 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

Cấu 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