

# HTML

## 2.1. Giới thiệu

## 2.2. Cấu trúc HTML

`<tagname attribute> content </tagname>`

tagname: tên thẻ

attribute: thuộc tính thẻ

## 2.3 Các thẻ HTML cơ bản

`<!DOCTYPE>`

`<html>`

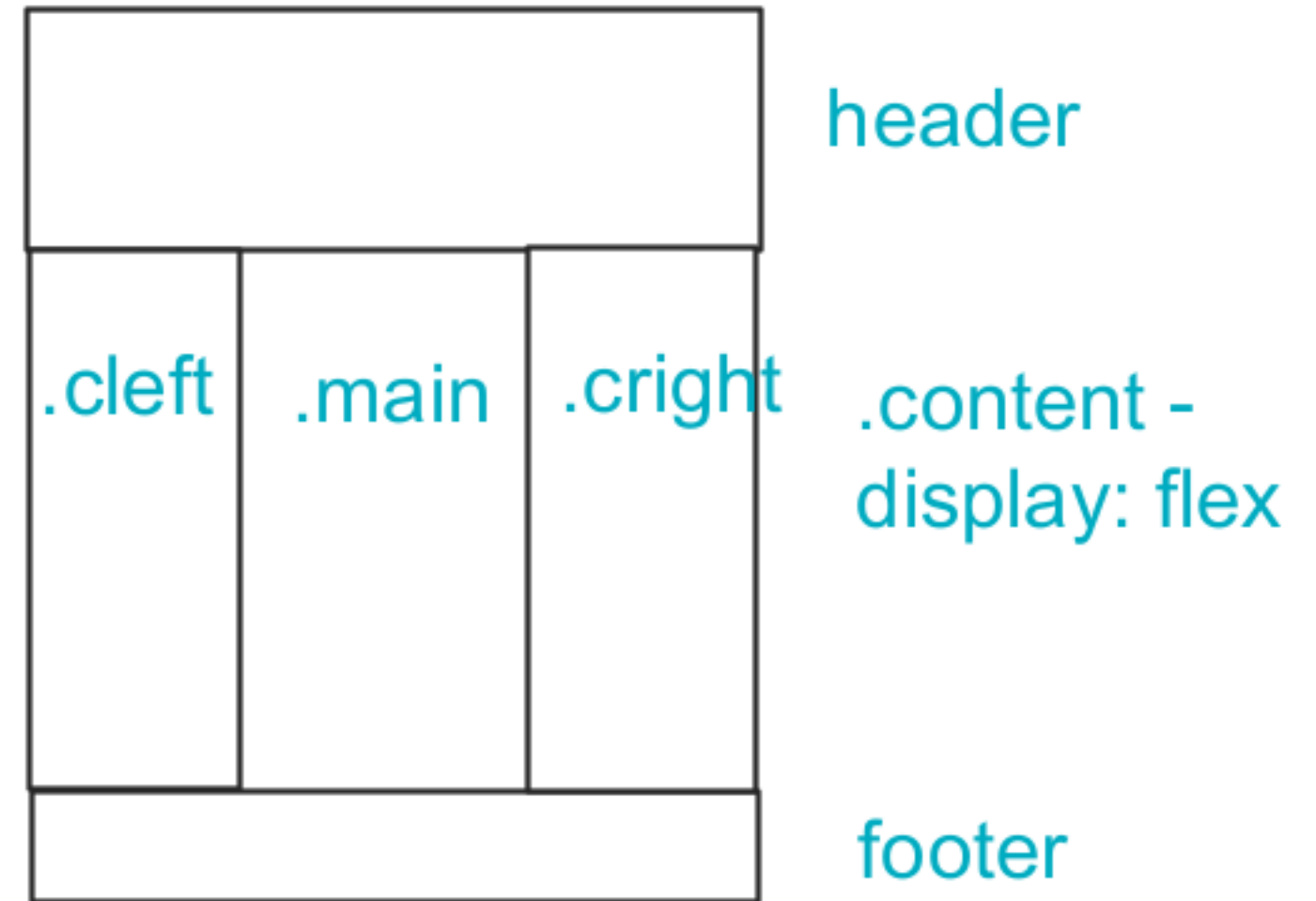
`<head>`

`<title>`

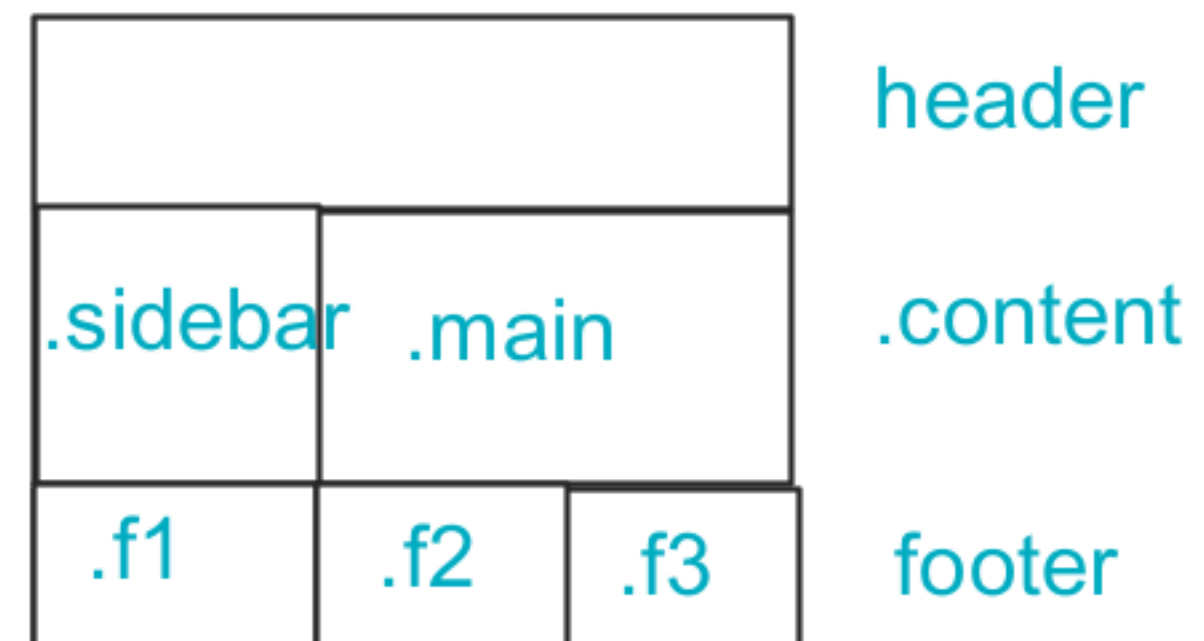
`<body>`

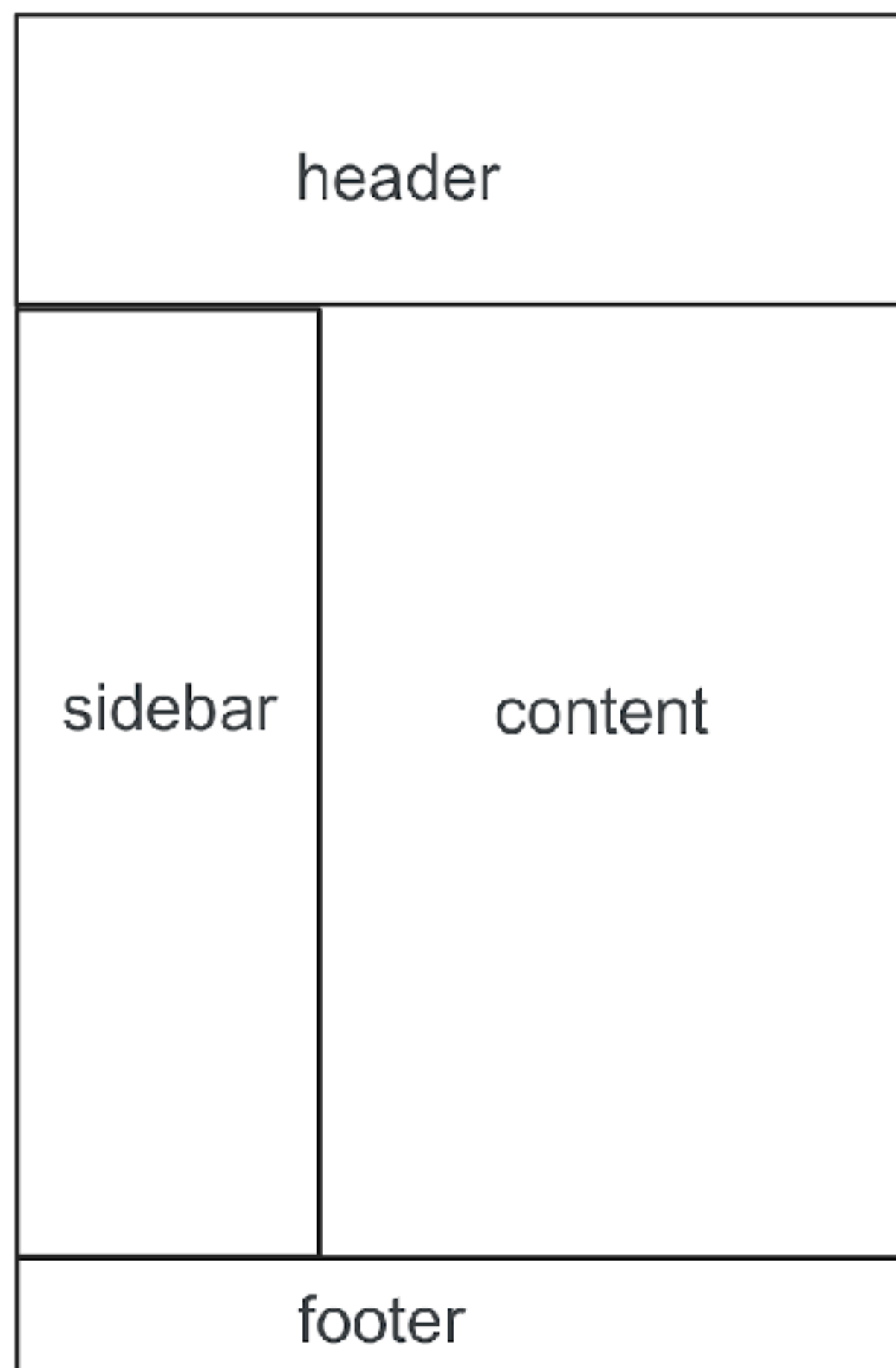
## 2.4 Form

### online TDC



### fit TDC





```
Welcome    index.html    demo.html X
demo.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-
6      <title>Layout</title>
7  </head>
8  <body>
9      <header>Header</header>
10     <div class="main">
11         <div class="sidebar">sidebar</div>
12         <div class="content">content</div>
13     </div>
14     <footer>footer</footer>
15 </body>
16 </html>
```

2.1 Tạo file index.html có giao diện như hình dưới, trong đó Home, Student information, Register là danh sách các liên kết:

- Home liên kết tới index.html
- Student information liên kết tới info.html
- Register liên kết tới register.html

- [Home](#)
- [Student information](#)
- [Register](#)

# WELCOME TO FIT

©2024 - FIT-TDC

2.2 Tạo file info.html chứa thông tin của SV (tuần 1)

2.3 Tạo trang register.html chứa thông tin như hình bên

- [Home](#)
- [Student information](#)
- [Register](#)

## Đăng ký tài khoản tìm việc làm thêm

First Name:	<input type="text" value="Enter your first name"/>
Last Name:	<input type="text" value="Enter your last name"/>
Email:	<input type="text" value="It should contain @"/>
Mobile:	<input type="text" value="only 10 digits are allowed"/>
Gender:	
Date Of Birth:	<input type="text" value="mm/dd/yyyy"/>
Address:	<input type="text"/>
City:	<input type="text"/>
Area PIN:	<input type="text"/>
State:	<input type="text"/>
Qualification:	<input type="text" value="Select Qualification"/>
Specialization:	<input type="checkbox"/> Computer Science <input type="checkbox"/> Information Technology <input type="checkbox"/> Computer Architecture <input type="checkbox"/> Tele Communication
Password:	<input type="password"/>
<input type="button" value="Registered"/>	

# CSS

## 2.5. Giới thiệu CSS

selector {attribute: value; attribute:value;..}

## 2.6. Phân loại CSS

- Internal CSS
- External CSS
- Inline CSS

## 2.7. CSS Selector

\*

tag

#

.

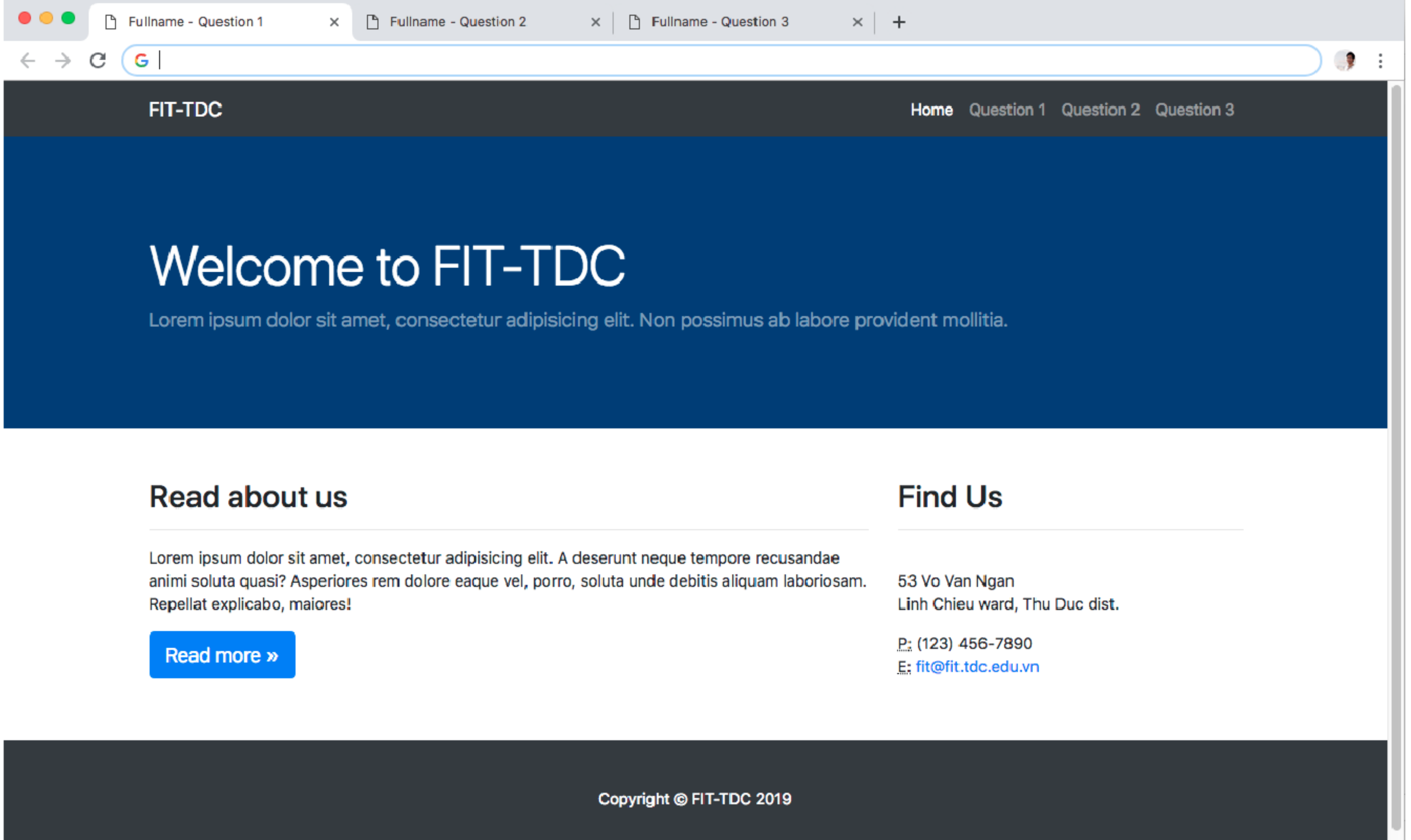
tag1 tag2

tag1>tag2

## 2.8. Các CSS cơ bản

- color
- background
- border
- padding
- margin
- ...

copyright - &copy;



# Welcome to FIT-TDC

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Non possimus ab labore provident mollitia.

## Read about us

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A deserunt neque tempore recusandae animi soluta quasi? Asperiores rem dolore eaque vel, porro, soluta unde debitis aliquam laboriosam. Repellat explicabo, maiores!

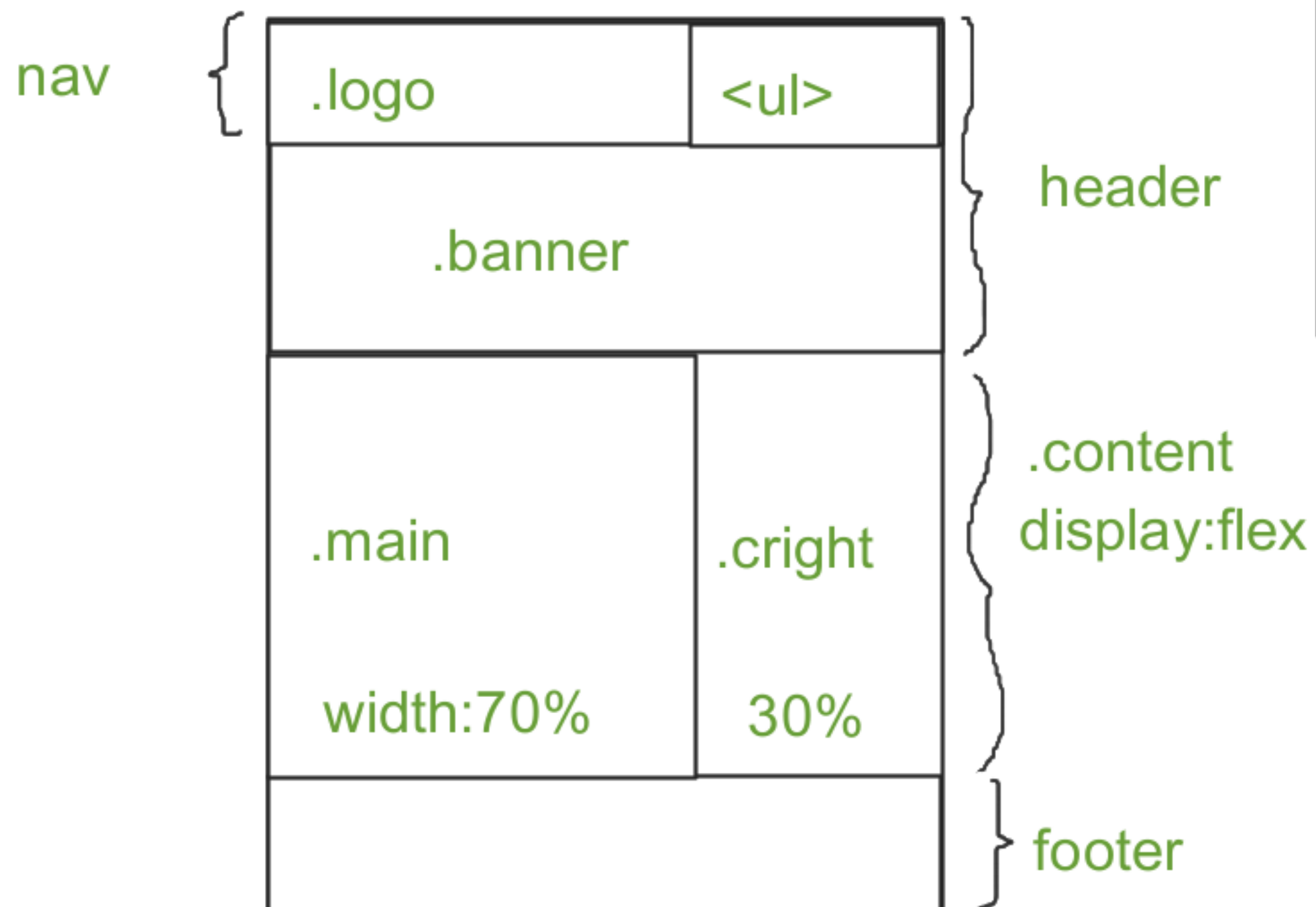
[Read more »](#)

## Find Us

53 Vo Van Ngan  
Linh Chieu ward, Thu Duc dist.

P: (123) 456-7890  
E: [fit@fit.tdc.edu.vn](mailto:fit@fit.tdc.edu.vn)

## layout fit tuần 3



nav

list-style:none

```
<ul>
  <li><a href='..'>Home</a></li>
  ...
</ul>
```

```
<a href='..'>
```

Readmore

background  
color  
padding  
border-radius  
text-decoration

bootstrap

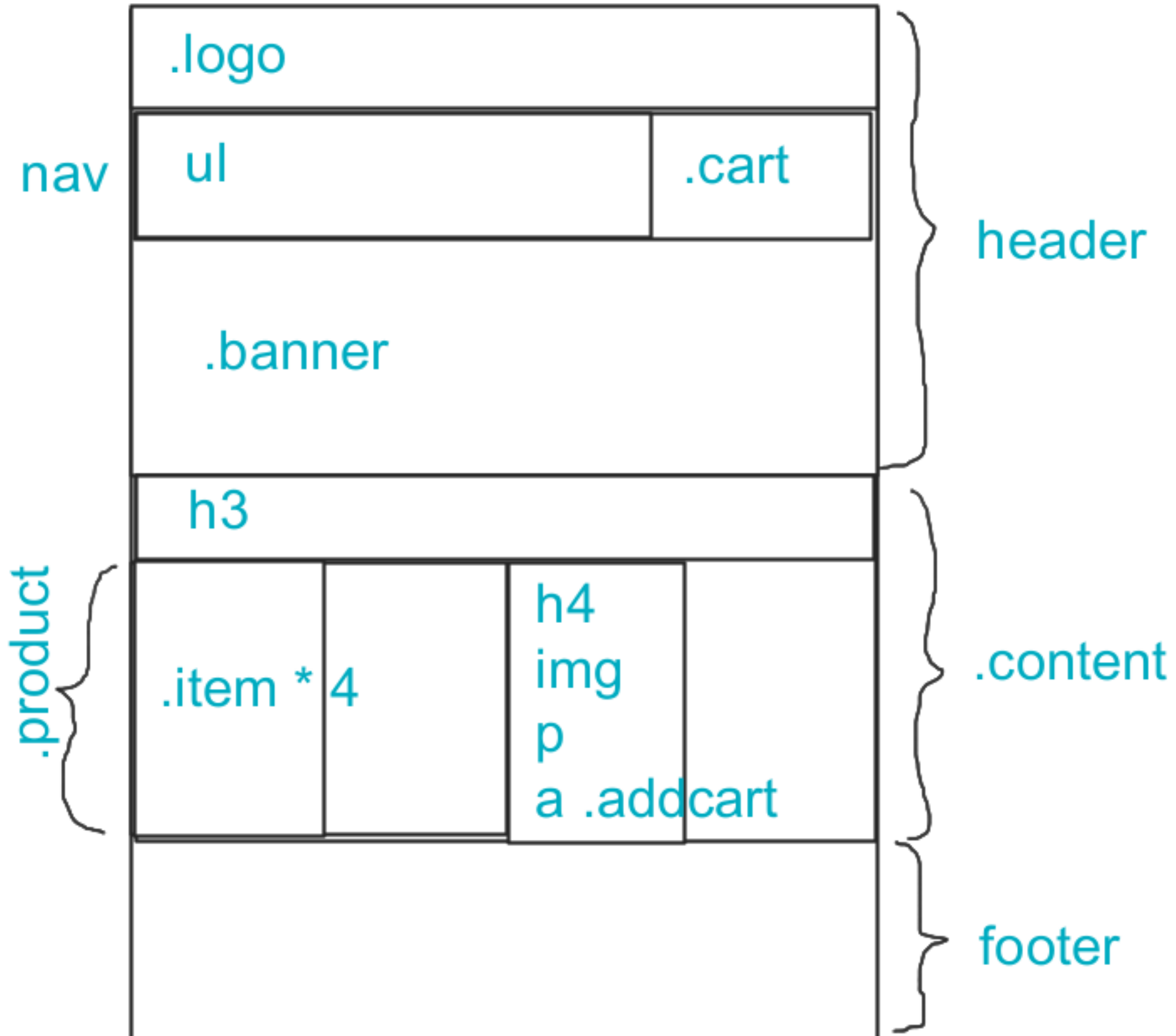
```
.container > .row > .col*12
```

.row



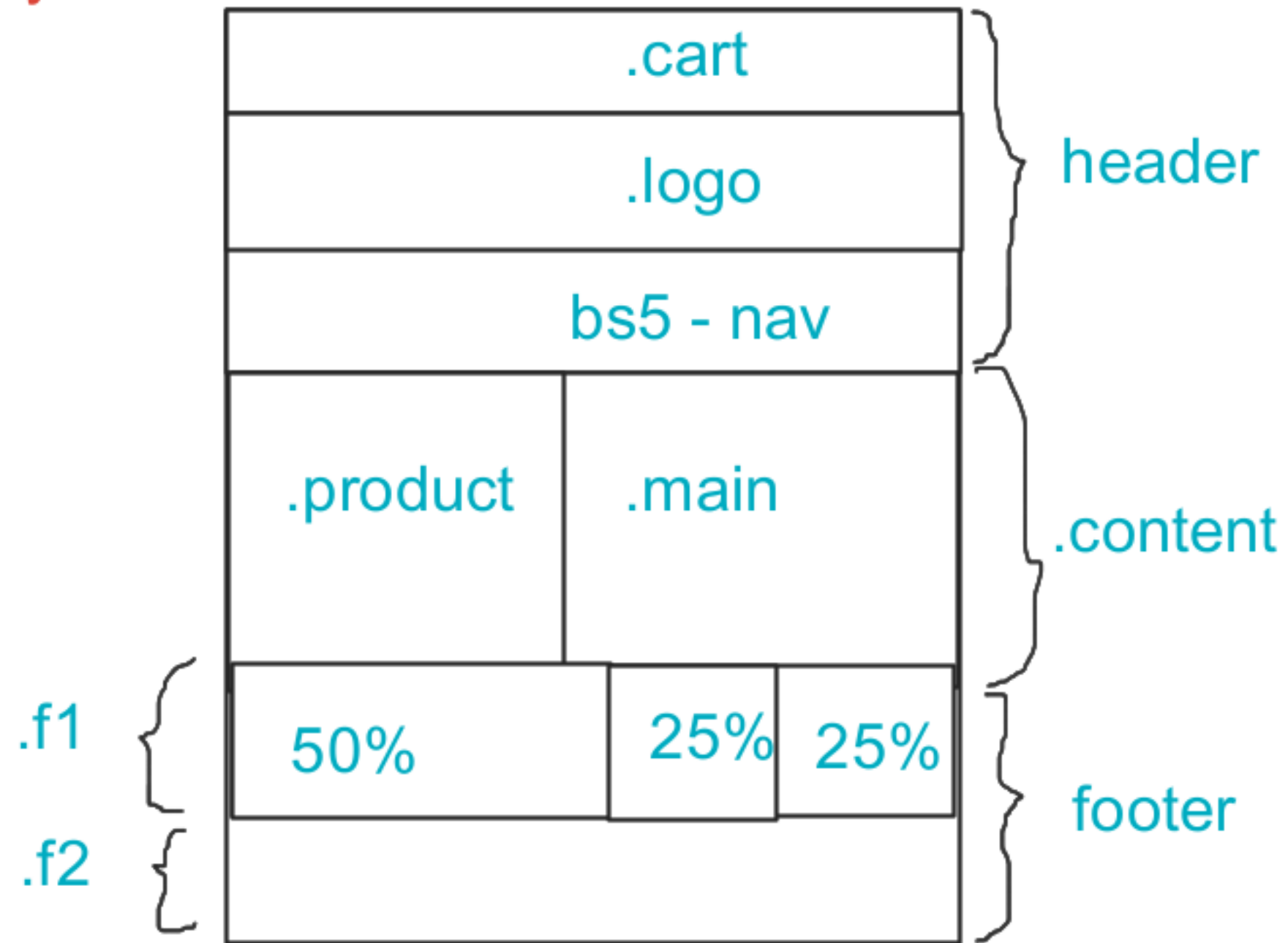


## Layout hitech tuan 4





layout watch

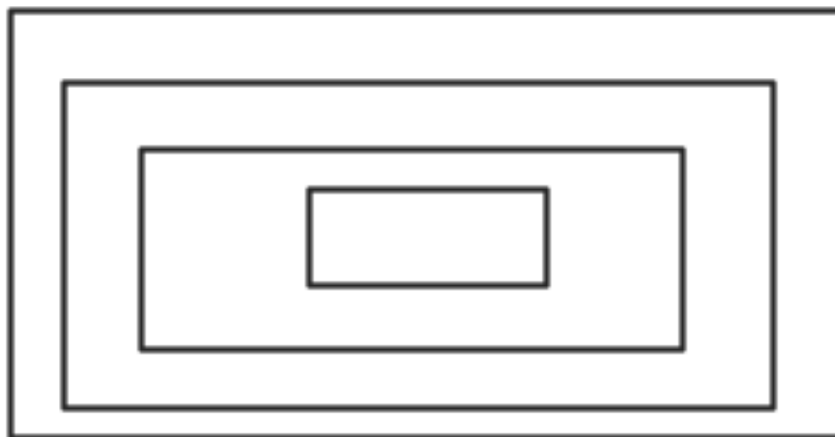


## CSS Pseudo-classes

`a:link`  
`a:visited`  
`a:hover`  
`a:active`

[https://  
www.w3schools.com/  
css/  
css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)

relative



1. Chọn đồ án: <https://themeforest.net/category/ecommerce>  
chụp hình desktop và mobile (dùng extension GoFullPage)

-> Link chọn đồ án: <https://forms.gle/GFKMjoJJqyJ6HENY7>

-> Kiểm tra đồ án chọn trùng:

<https://docs.google.com/spreadsheets/d/1BLvKJ2oEI9vPb1PnDg6P3gF59bKt1tfP9BXzIVyWHQA/edit?resourcekey#gid=1062500864>

2. Nộp đồ án trên el gồm:

```
projectname
|--public
|   |--css
|   |--js
|   |--images
|   |--...
|--mockups
|   |--desktop.jpg
|   |--mobile.jpg
|--index.html
```

# Javascript

## 1. Cách đặt:

```
<script>  
  //code js here  
</script>
```

1

```
<script src="script.js">  
</script>
```

2

## 2. Biến:

```
let varName;  
let varName = value;
```

## 3. Mảng

```
let arrName = [item1, item2, ...];
```

arrName.length -> lấy chiều dài mảng

## Duyệt mảng dùng forEach

```
arrName.forEach(function(item, index, array) {  
  // ... do something with item  
});
```

## 4. Hằng:

```
const consName = value;
```

## 5. Function:

```
function functionName (parameter, parameter =  
defaultvalue) {  
  // code js here  
}
```

```
let functionName = function(parameter, parameter  
= defaultvalue) {  
  // code js here  
}
```

## 5. Function:

arrow function

```
let functionName = (parameter, parameter = defaultvalue) => {  
  //code js here  
}
```

## 6. Event:

Mouse: click, contextmenu, mouseover/mouseout, mousedown/mouseup, mousemove, ..

Keyboard: keydown, keyup, ..

Form: submit, focus, ..

**addEventListener**

```
element.addEventListener(event, handler, [options]);
```

event: Event name, e.g. "click"

handler: The handler function

options: An additional optional object with properties: once, capture, passive

## Bai 1 - JS

1. Lấy 3 thành phần: operation, result, button
2. Duyệt mảng button: Bắt sự kiện click chuột cho item, và kiểm tra
  - Nếu giá trị button là "Enter" -> tính operation (eval(1+1)->2)
  - Nếu giá trị button là "C" -> xóa nội dung operation và result
  - Ngược lại thêm vào operation

## Bai 2 - JS

1. Viết hàm trả về giờ : phút : giây hiện hành và gắn vào clock
2. Gắn hàm vào setInterval (function,delay - millisecond)

```
let count = 1;  
function plusOne(){  
    alert(count);  
    return count++;  
}  
let t = setInterval(plusOne,1000);  
clearInterval(t)
```



## Bai 3 - JS

1. Viết hàm chong chóng xoay - transform: rotate(90deg)
2. Khi ấn start -> setInterval
3. Khi ấn stop -> clearInterval

## Bai 8 - JS

1. Lấy thành phần hình to và hình nhỏ
2. Duyệt mảng hình nhỏ, sau đó bắt sự kiện click chuột  
-> Lấy giá trị thuộc tính src của hình nhỏ tương ứng gán cho thuộc tính src của hình to

## Bai 10 - JS

1. Lấy thành phần chậu hoa và cái đĩa
  2. Duyệt mảng chậu hoa, bắt sự kiện click chuột từng chậu hoa
    - Khi click vào item, lưu thông tin: tên hình chậu hoa (f1/f2/f3) và giá tiền (200/300/500)
  3. Duyệt mảng cái đĩa, bắt sự kiện click chuột
- Kiểm tra: - Đã chọn hoa hay chưa?  
- Chậu đã có hoa hay chưa?  
- Kiểm tra xem đủ tiền hay ko?  
- Ngược lại, gắn hoa vào chậu

<https://javascript.info/dom-navigation>

## Bai 5

<https://javascript.info/modifying-document>

1. Tạo 100 hình tròn
2. Viết hàm ngẫu nhiên
3. Viết hàm đổi màu ngẫu nhiên cho 100 hình tròn

## Bai 6

1. Viết hàm ngẫu nhiên
2. Viết hàm changePosition cho 10 hình tròn (css 2 thuộc tính: top, left)
3. Khi click chuột nút start, setInterval(changePosition,500), và setTimeout(stopGame,10000)
4. Lấy 10 hình tròn, duyệt và bắt sự kiện click chuột
  - Tăng score lên
5. Viết hàm stopGame();-> clearInterval và xuất class timeout (display:block)

## Bai 7

1. Viết hàm ngẫu nhiên
2. Tạo 100 số (number)
3. Viết hàm changePosition()
4. Khi bấm start -> changePosition, và setTimeout(stopGame,10000)
5. Lấy 100 số, duyệt và bắt sự kiện click chuột
  - Kiểm tra số tăng theo đúng thứ tự -> Tạo biến đếm để so sánh
6. Viết hàm stopGame();



## Bai 9

1. Khi click chuột vào nút send, nội dung #message chèn vào .chat
2. Khi bấm enter, nội dung #message chèn vào .chat
3. autoscroll
4. Hide/show chatbox ( transition css, height:0)

<https://javascript.info/keyboard-events>

## Form Validation

1. Kiểm tra trường bắt buộc nhập `/^.+$/`
2. Kiểm tra trường bắt buộc nhập 1 hay nhiều số `/^\d+$/` or `/^[0-9]+$/`
3. Kiểm tra trường MSSV TDC - 21211TT1234 `/^\d{5}[a-zA-Z]{2}\d{4}$/`
4. Kiểm tra username có ít nhất 6 ký tự word `/^\w{6,}$/`
5. Kiểm tra field là email - thao123@mail.com  
`/^\w+@[a-z]+\.[a-z]+$/`
6. Tối thiểu 8 ký tự, trong đó có ít nhất một ký tự đặc biệt  
`/^(?=.*\W).\{8,}$/`

## Form Validation

### Các bước kiểm tra form

1. Thêm thuộc tính novalidate vào thẻ form  
`<form class="row" novalidate>`
2. Để thông báo lỗi, thêm class invalid-feedback ngay bên dưới <input> như sau:  
`<div class="invalid-feedback">`  
    Sai username  
`</div>`
3. Thêm 2 thuộc tính required và pattern cho từng trường input (\*)
4. Viết hàm kiểm tra  

```
const form = document.querySelector('form');
form.addEventListener("submit", (e) => {
  if (form.checkValidity() === false) {
    //Ngăn ko cho form được gửi đi
    e.preventDefault();
    e.stopPropagation()
  }
  form.classList.add('was-validated');
});
```

1. Viết hàm ngẫu nhiên
2. Viết hàm tạo phép toán ngẫu nhiên -> trả về kết quả đúng (return eval(phép tính))
3. Viết hàm tạo 10 hình tròn createCircle
4. Viết hàm thay đổi giá trị 10 hình tròn
  - hình tròn đầu tiên mang giá trị đúng
  - 9 hình tròn sau mang giá trị ngẫu nhiên -99 -> 198 (có thể kiểm tra khác đáp án)
5. Viết hàm thay đổi vị trí changePosition
6. Bắt sự kiện bấm nút start
  - Gọi hàm 2
  - Gọi hàm 3
  - Gọi hàm 4
  - Gọi hàm 5
  - Bắt sự kiện click chuột cho 10 circle
    - Kiểm tra nếu đúng -> tăng điểm
    - Ngược lại -> giảm điểm
  - Gọi Hàm 2
  - Gọi Hàm 4
  - Gọi Hàm 5

```
let x;
let y;
let o = ['+', '-'];
let op = o[NN(0,1)];
x + op + y
```

1. Lấy 3 đối tượng #color, .btn-stitch, và .box

2. Bắt sự kiện input cho color -> lấy màu

```
let c="#000";
color.addEventListener("input/change",function(){
  c = this.value;
})
```

3. Duyệt và bắt sự kiện click chuột cho .btn-stitch -> lấy biểu tượng

```
let icon="&times;";
stitch.forEach(function(item){
  item.addEventListener("click",function(){
    icon = item.textContent;
  })
})
```

4. Duyệt và bắt sự kiện click chuột cho .box -> gắn biểu tượng và màu vào box

```
box.forEach(function(item){
  item.addEventListener("click",function(){
    item.innerHTML = "<div class='stitch'>" + icon + "</div>";
    item.style.color = c;
  })
})
```