**PHIẾU HỌC TẬP CSS CƠ BẢN**

**Họ và tên:** Hà Văn Tâm

**Mã sinh viên:** 2451271161

**Lớp:**66ANM1

**Ngày làm:**23/2/2026

**PHIẾU HỌC TẬP --- CÚ PHÁP CSS & BỘ CHỌN (SELECTORS)**

**Tài liệu đã xem (Link):**

**1. Kiến thức cốt lõi (Tóm tắt bằng lời của bạn)**

**1.1. CSS là gì? (Viết bằng lời riêng của bạn)**

- Css là ngôn ngữ dùng để thêm màu sắc, trang trí cho trang web

CSS là viết tắt của

- Cascading Style Sheets

Nó có vai trò gì trong 1 trang web?

* HTML chịu trách nhiệm: Là 1 khung xương của 1 trang web
* CSS chịu trách nhiệm: Làm cho trang web thêm màu sắc và dễ nhìn, bố cục, giao diện
* JavaScript chịu trách nhiệm: Xử lý tương tác và hành vi động

**1.2. Cú pháp cơ bản của CSS**

Một quy tắc CSS (CSS Rule) gồm 2 phần:

selector {  
 property: value;  
}

**Giải thích:**

* **Selector** (bộ chọn): Chọn phần tử HTML cần áp dụng CSS
* **Property** (thuộc tính): Đặc điểm muốn thay đổi (color, width, margin)
* **Value** (giá trị): Giá trị của thuộc tính

**1.3. Ba cách áp dụng CSS vào HTML**

Hoàn thành bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cách áp dụng** | **Vị trí code** | **Ưu điểm** | **Nhược điểm** |
| **Inline CSS** | *Trong thẻ Html*  *(style=””)* | *Nhạm, đơn giản* | *Khó quản lý, không tái*  *sử dụng* |
| **Internal CSS** | *Trong <style> ở*  *<head>* | *Gọn trong 1 file* | *Không tái sử dụng nhiều*  *trang* |
| **External CSS** | *File .css riêng* | *Chuyên nghiệp, tái sử*  *dụng* | *Cần liên kết* |

**2. Các loại Selector cơ bản**

Hoàn thành bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại Selector** | **Cú pháp** | **Ví dụ code** | **Ý nghĩa** |
| **Universal Selector** | \* | \* { margin: 0; } | *Chọn tất cả phần tử* |
| **Type Selector** | element | p { color: blue; } | *Chọn theo tên thẻ* |
| **Class Selector** | .className | .btn { padding: 10px; } | *Chọn theo class* |
| **ID Selector** | #idName | #header { background: gray; } | *Chọn theo id* |
| **Descendant Selector** | parent child | nav a { text-decoration: none; } | *Chọn p bên trong div* |
| **Group Selector** | selector1, selector2 | h1, h2, h3 { font-family: Arial; } | *Chọn nhiều selector* |
| **Pseudo-class Selector** | selector:state | a:hover { color: red; } | *(Tự điền)* |

**3. Bài tập thực hành 1 (20 phút): Áp dụng CSS & Selector**

**Mục tiêu:**

* Tạo file style.css và liên kết vào HTML
* Dùng các loại selector: type, class, id, group, descendant, pseudo-class
* Thực hành styling cơ bản cho menu và header

**Cho trước file HTML (không sửa cấu trúc):**

<!DOCTYPE html>  
<html lang="vi">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Lab CSS Cơ Bản</title>  
 <!-- TODO: Liên kết file style.css vào đây -->  
</head>  
<body>  
 <header id="main-header">  
 <h1>My Blog</h1>  
 </header>  
   
 <nav>  
 <a href="#" class="nav-link">Home</a>  
 <a href="#" class="nav-link">About</a>  
 <a href="#" class="nav-link">Contact</a>  
 </nav>  
   
 <main>  
 <section class="intro">  
 <h2>Chào mừng đến với Blog</h2>  
 <p>Đây là trang blog về lập trình web.</p>  
 </section>  
 </main>  
   
 <footer>  
 <p>&copy; 2026 My Blog. All rights reserved.</p>  
 </footer>  
</body>  
</html>

**Yêu cầu CSS (viết vào file style.css):**

**Bước 1:** Liên kết file CSS vào HTML bằng thẻ <link> trong <head>.

<link ref=”stylesheet” href=”style.css”>

**Bước 2:** Viết CSS cho các yêu cầu sau:

/\* TODO 1: Dùng universal selector để reset margin và padding về 0 cho toàn trang \*/  
{

margin: 0;

padding: 0;

box-sizing: border-box;  
  
/\* TODO 2: Dùng type selector cho body \*/  
/\* Yêu cầu: font-family: Arial; background-color: #f5f5f5; \*/

html, body {

  height: 100%;

}

body {

  font-family: Arial, sans-serif;

  background-color: #f5f5f5;

  display: flex;

  flex-direction: column;

}

/\* TODO 3: Dùng ID selector cho #main-header \*/  
/\* Yêu cầu: background-color: #333; color: white; text-align: center; padding: 20px; \*/  
#main-header {

  background-color: #333;

  color: white;

  text-align: center;

  padding: 20px;

}

/\* Menu container \*/

nav {

  background: #fff;

  text-align: center;

  padding: 12px 0;

  border-bottom: 1px solid #e5e5e5;

}

/\* TODO 4: Dùng class selector cho .nav-link \*/  
/\* Yêu cầu: color: #333; text-decoration: none; padding: 10px 15px; \*/  
.nav-link{

color: #333;

text-decoration:none;

padding: 10px 15px;

border-radius: 8px;

transition: 0.2s;

}  
  
/\* TODO 5: Dùng group selector cho h1, h2, h3 \*/  
/\* Yêu cầu: font-family: 'Georgia', serif; \*/(Do front Georgia bị lỗi chữ “ế” nên e thay bằng time new roman)

h1, h2, h3 {

  font-family: "Times New Roman", Times, serif;}

/\* TODO 6: Dùng descendant selector cho nav a \*/  
/\* Yêu cầu: display: inline-block; margin: 0 10px; \*/  
nav a{

display: inline-block;

margin: 0 10px;

}  
  
/\* TODO 7: Dùng pseudo-class selector cho a:hover \*/  
/\* Yêu cầu: color: #ff6600; background-color: #eee; \*/  
a:hover{

color: #ff6600;

background-color: #eee;

}

**Kết quả mong đợi:**

1. Menu nằm ngang, các link cách đều nhau
2. Khi rê chuột vào link, màu chữ và nền thay đổi
3. Header có nền tối, chữ trắng, căn giữa

**Ảnh kết quả (Dán ảnh trình duyệt vào đây):**

*(Dán ảnh 1: Giao diện bình thường)*

*(Dán ảnh 2: Khi hover vào link)*

**PHIẾU HỌC TẬP --- BOX MODEL & BỐ CỤC CƠ BẢN**

**Tài liệu đã xem (Link):**

**1. Kiến thức cốt lõi: CSS Box Model**

**1.1. Box Model gồm những thành phần gì?**

Vẽ sơ đồ Box Model (từ trong ra ngoài):

┌─────────────────────────────────────┐  
│ MARGIN (lề ngoài) │  
│ ┌───────────────────────────────┐ │  
│ │ BORDER (đường viền) │ │  
│ │ ┌─────────────────────────┐ │ │  
│ │ │ PADDING (lề trong) │ │ │  
│ │ │ ┌───────────────────┐ │ │ │  
│ │ │ │ CONTENT │ │ │ │  
│ │ │ │ (Nội dung) │ │ │ │  
│ │ │ └───────────────────┘ │ │ │  
│ │ └─────────────────────────┘ │ │  
│ └───────────────────────────────┘ │  
└─────────────────────────────────────┘

**Giải thích:**

* **Content**: Là phần nội dung thực tế ( hình ảnh, văn bản hoặc video)
* **Padding**: Là khoảng cách giữa content và border, nó làm nội dung thoáng hơn
* **Border**: Là đường viền bao quanh padding và content
* **Margin**: Là khoảng cách bên ngoài phần tử, tạo khoảng cách giữa các phần tử với nhau

**1.2. Hai cách tính kích thước Box**

Hoàn thành bảng:

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Cách tính width** | **Ưu/Nhược điểm** |
| box-sizing: content-box; | *Tổng width=width + padding*  *+ border* | *Khó kiểm soát kích thước thật*  *của phần tử* |
| box-sizing: border-box; | *Tổng width = width (đã bao*  *gồm padding \_ border)* | *Dễ kiểm soát layout, tính toán*  *chính xác* |

**Câu hỏi:** Nên dùng box-sizing nào cho toàn bộ trang? Tại sao?

Nên dùng: box-sizing: border-box;

Vì:

 Dễ tính toán kích thước

 Không bị “phình” layout ngoài ý muốn

 Phù hợp khi làm responsive

 Được dùng phổ biến trong thực tế

**2. Bài tập thực hành 2 (35 phút): Box Model & Layout Card**

**Mục tiêu:**

* Tạo 3 card bài viết với box model rõ ràng
* Dùng margin, padding, border để tách biệt các card
* Áp dụng box-sizing: border-box
* Bố trí card theo hàng ngang (dùng display: inline-block hoặc float)

**Thêm HTML sau vào trong <main> (sau phần intro):**

<section class="posts">  
 <article class="card">  
 <h3>Bài viết 1</h3>  
 <p>Mô tả ngắn về bài viết 1.</p>  
 <a href="#" class="btn">Đọc thêm</a>  
 </article>  
   
 <article class="card">  
 <h3>Bài viết 2</h3>  
 <p>Mô tả ngắn về bài viết 2.</p>  
 <a href="#" class="btn">Đọc thêm</a>  
 </article>  
   
 <article class="card">  
 <h3>Bài viết 3</h3>  
 <p>Mô tả ngắn về bài viết 3.</p>  
 <a href="#" class="btn">Đọc thêm</a>  
 </article>  
</section>

**Yêu cầu CSS:**

/\* Bước 1: Reset box-sizing cho toàn trang \*/  
\* {  
 box-sizing: border-box;  
}  
  
/\* Bước 2: Style cho .posts container \*/  
.posts {  
 /\* TODO: Căn giữa, margin trên dưới, max-width \*/

max-width: 1200px;

    margin: 40px auto;

    text-align: center;

    font-size: 0;

}  
  
/\* Bước 3: Style cho .card \*/  
.card {

    width: calc(33.33% - 20px);

    display: inline-block;

    font-size: 16px;

    vertical-align: top;

    border: 1px solid #ddd;

    padding: 20px;

    margin: 10px;

    background: white;

    border-radius: 8px;

}

/\* Bước 4: Style cho .btn \*/  
.btn {

    display: inline-block;

    padding: 8px 16px;

    background: #007bff;

    color: white;

    text-decoration: none;

    border-radius: 4px;

}

.btn:hover {

    background: #0056b3;

}

**Kết quả mong đợi:**

1. Ba card nằm ngang trên 1 hàng
2. Mỗi card có viền, padding, margin rõ ràng
3. Nút "Đọc thêm" có màu xanh, hover đổi màu đậm hơn

**Ảnh kết quả (Dán ảnh vào đây):**

*(Dán ảnh: Giao diện 3 card)*

**PHIẾU HỌC TẬP --- FLEXBOX & CSS GRID**

**Tài liệu đã xem (Link):**

**1. Kiến thức cốt lõi: So sánh Flexbox và Grid**

Hoàn thành bảng sau:

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Flexbox** | **CSS Grid** |
| **Bố cục** | *1 chiều(hàng hoặc cột)* | *2 chiều (hàng và cột)* |
| **Phù hợp với** | *Menu,thanh điều hướng*  *card 1 hàng* | *Layout trang phức tạp nhiều*  *hàng nhiều cột* |
| **Thuộc tính chính** | display: flex;  justify-content  align-items | display: grid;  grid-template-columns  grid-template-rows |
| **Ví dụ sử dụng** | *Menu ngang, navbar* | *Gallery, layout trang báo* |

**1.1. Flexbox - Các thuộc tính quan trọng**

.flex-container {  
 display: flex;  
   
 /\* Căn chỉnh theo trục ngang (main axis) \*/  
 justify-content: \_\_\_\_\_\_\_\_\_\_\_; /\* flex-start | center | space-between | space-around \*/  
   
 /\* Căn chỉnh theo trục dọc (cross axis) \*/  
 align-items: \_\_\_\_\_\_\_\_\_\_\_; /\* flex-start | center | flex-end | stretch \*/  
   
 /\* Cho phép xuống hàng \*/  
 flex-wrap: \_\_\_\_\_\_\_\_\_\_\_; /\* nowrap | wrap \*/  
   
 /\* Khoảng cách giữa các item \*/  
 gap: \_\_\_\_\_\_\_\_\_\_\_;  
}

**1.2. CSS Grid - Các thuộc tính quan trọng**

.grid-container {  
 display: grid;  
   
 /\* Định nghĩa số cột và kích thước \*/  
 grid-template-columns: \_\_\_\_\_\_\_\_\_\_\_; /\* repeat(3, 1fr) | 200px 1fr 2fr \*/  
   
 /\* Định nghĩa số hàng và kích thước \*/  
 grid-template-rows: \_\_\_\_\_\_\_\_\_\_\_;  
   
 /\* Khoảng cách giữa các ô \*/  
 gap: \_\_\_\_\_\_\_\_\_\_\_;  
}

**2. Bài tập thực hành 3A (20 phút): Flexbox - Menu & Card Layout**

**Mục tiêu:**

* Dùng Flexbox cho menu ngang
* Dùng Flexbox cho layout card (thay thế inline-block)
* Responsive: card xuống 1 cột khi màn hình nhỏ

**Yêu cầu CSS (cải tiến code cũ):**

/\* Cải tiến nav bằng Flexbox \*/  
nav {  
 /\* TODO:  
 display: flex  
 justify-content: center  
 align-items: center  
 gap: 20px  
 background: #f8f9fa  
 padding: 15px  
 \*/  
}  
  
/\* Cải tiến .posts bằng Flexbox \*/  
.posts {  
 /\* TODO:  
 - display: flex  
 - flex-wrap: wrap  
 - gap: 20px  
 - justify-content: center (hoặc space-between)  
 - max-width: 1200px  
 - margin: 40px auto  
 \*/  
}  
  
.card {  
 /\* TODO:  
 - flex: 0 0 calc(33.33% - 20px)  
 - (hoặc flex: 1 1 300px cho responsive tự động)  
 \*/  
}  
  
/\* Responsive: Khi màn hình < 768px, card chiếm full width \*/  
@media (max-width: 768px) {  
 .posts {  
 /\* TODO: flex-direction: column \*/  
 }  
   
 .card {  
 /\* TODO: flex: 1 1 100% \*/  
 }  
}

**Kết quả mong đợi:**

1. Menu các link nằm ngang, căn giữa, cách đều
2. Card layout linh hoạt, tự động xuống hàng khi hết chỗ
3. Khi thu nhỏ màn hình < 768px, card xếp thành 1 cột

**Ảnh kết quả (Dán ảnh vào đây):**

*(Dán ảnh 1: Desktop view - 3 card ngang)*

*(Dán ảnh 2: Mobile view - card xếp dọc)*

**3. Bài tập thực hành 3B (20 phút): CSS Grid - Gallery Layout**

**Mục tiêu:**

* Tạo lưới 2 hàng x 3 cột bằng CSS Grid
* Responsive: tự động giảm số cột khi màn hình nhỏ

**Thêm HTML sau vào cuối <main>:**

<section class="gallery">  
 <div class="item">1</div>  
 <div class="item">2</div>  
 <div class="item">3</div>  
 <div class="item">4</div>  
 <div class="item">5</div>  
 <div class="item">6</div>  
</section>

**Yêu cầu CSS:**

.gallery {  
 /\* TODO:  
 - display: grid  
 - grid-template-columns: repeat(3, 1fr) ← 3 cột bằng nhau  
 - gap: 16px  
 - max-width: 1200px  
 - margin: 40px auto  
 \*/  
}  
  
.item {  
 /\* TODO:  
 - background: linear-gradient(135deg, #667eea 0%, #764ba2 100%)  
 - color: white  
 - display: flex  
 - justify-content: center  
 - align-items: center  
 - height: 150px  
 - font-size: 2rem  
 - border-radius: 8px  
 \*/  
}  
  
/\* Responsive: Tự động giảm cột khi màn hình nhỏ \*/  
.gallery {  
 /\* TODO: Dùng auto-fit/auto-fill \*/  
 grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
}

**Giải thích kỹ thuật:**

**repeat(auto-fit, minmax(200px, 1fr))** có nghĩa gì?

* repeat(): Hàm lặp để tạo nhiều cột/ hàng theo cùng 1 mẫu
* auto-fit: Tự động đẩy các cột, nếu thiếu iteam thì các cột trống sẽ co lại
* minmax(200px, 1fr): cột nhỏ nhất 200px, lớn nhất giãn đều theo phần còn lại(1fr)

**Kết quả mong đợi:**

1. Desktop: 3 cột, 2 hàng (6 ô)
2. Tablet: 2 cột, 3 hàng
3. Mobile: 1 cột, 6 hàng

**Ảnh kết quả (Dán ảnh vào đây):**

*(Dán ảnh 1: Desktop - 3 cột)*

*(Dán ảnh 2: Mobile - 1 cột)*

**PHIẾU HỌC TẬP --- ĐỊNH VỊ CSS & RESPONSIVE DESIGN**[**[2]**](#fn2)

**Tài liệu đã xem (Link):**

**1. Kiến thức cốt lõi: CSS Position**

Hoàn thành bảng:

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Ý nghĩa** | **Ví dụ sử dụng** |
| static | *Giá trị mặc định, phần tử hiển thị theo*  *luồng bình thường của trang, không dùng*  *top, left, right, bottom* | *Đoạn văn bản bình thường* |
| relative | *Định vị tương đối so với vị trí ban đầu của*  *chính nó. Vẫn chiếm chỗ trong layout* | *Dịch chuyển nhẹ 1 icon* |
| absolute | *Định vị tuyệt đối so với phần tử cha gần nhất*  *có position khác static. Không chiếm chỗ trong*  *layout* | *Tooltip, badge, popup nhỏ* |
| fixed | *Định vị cố định so với viewport (màn hình). Không*  *bị cuộn theo trang* | *Nút “Back to top”* |
| sticky | *Kết hợp relative + fixed. Khi cuộn tới vị trí xác định*  *thì nó dính lại* | *Thanh menu dính trên đầu trang* |

**2. Responsive Design - Media Queries**

**Cú pháp cơ bản:**

/\* Desktop First \*/  
@media (max-width: 768px) {  
 /\* Code CSS cho màn hình nhỏ hơn 768px \*/  
}  
  
/\* Mobile First \*/  
@media (min-width: 768px) {  
 /\* Code CSS cho màn hình lớn hơn 768px \*/  
}

**Breakpoints phổ biến:**

* **Mobile**: < 576px
* **Tablet**: 576px - 768px
* **Desktop**: > 768px
* **Large Desktop**: > 1200px

**3. Bài tập thực hành 4 (30 phút): Tạo Header Sticky + Responsive**

**Yêu cầu:**

1. Header cố định trên cùng khi scroll (sticky)
2. Thêm nút "Scroll to Top" (fixed, góc dưới bên phải)
3. Responsive: Menu chuyển icon burger khi mobile (chỉ làm style, không cần JavaScript)

**Cập nhật HTML:**

<header id="main-header" class="sticky-header">  
 <h1>My Blog</h1>  
</header>  
  
<!-- Nút Scroll to Top -->  
<a href="#" class="scroll-top">↑</a>

**Yêu cầu CSS:**

/\* Header sticky \*/  
.sticky-header {  
 position: sticky;  
 top: 0;  
 z-index: 100;  
 /\* TODO: Thêm box-shadow để tạo hiệu ứng nổi \*/  
}  
  
/\* Nút Scroll to Top \*/  
.scroll-top {  
 position: fixed;  
 bottom: 30px;  
 right: 30px;  
 /\* TODO:  
 - width: 50px, height: 50px  
 - background: #007bff  
 - color: white  
 - border-radius: 50%  
 - display: flex, justify/align center  
 - text-decoration: none  
 - font-size: 24px  
 \*/  
}  
  
.scroll-top:hover {  
 /\* TODO: background: #0056b3, transform: scale(1.1) \*/  
}  
  
/\* Responsive \*/  
@media (max-width: 768px) {  
 nav {  
 /\* TODO: flex-direction: column \*/  
 }  
}

**Kết quả mong đợi:**

1. Header luôn cố định trên cùng khi scroll
2. Nút "↑" ở góc dưới bên phải, hover phóng to
3. Mobile: menu xếp dọc

**Ảnh kết quả (Dán ảnh vào đây):**

*(Dán ảnh: Giao diện có header sticky và nút scroll)*

**MINI TEST - Tự Kiểm Tra**

**Phần 1: Selector & Cú pháp**

**1. Selector nào chọn tất cả thẻ <p> nằm trong <div>?**

* A. div p
* B. div > p
* C. div + p

**2. Pseudo-class :hover dùng để?**

* A. Chọn phần tử đầu tiên
* B. Chọn phần tử khi rê chuột
* C. Chọn phần tử cuối cùng

**Phần 2: Box Model**

**3. box-sizing: border-box có nghĩa?**

* A. Width bao gồm content + padding + border
* B. Width chỉ tính content
* C. Width tính cả margin

**4. Thứ tự từ trong ra ngoài của Box Model?**

* A. Content → Padding → Border → Margin
* B. Content → Border → Padding → Margin
* C. Content → Margin → Padding → Border

**Phần 3: Flexbox & Grid**

**5. Flexbox phù hợp với layout?**

* A. 1 chiều (hàng hoặc cột)
* B. 2 chiều (hàng và cột)
* C. Không phù hợp với layout

**6. justify-content: space-between có tác dụng?**

* A. Căn giữa các item
* B. Chia đều khoảng trống giữa các item
* C. Chia đều khoảng trống cả 2 bên

**7. Grid tốt hơn Flexbox khi nào?**

* A. Menu ngang đơn giản
* B. Layout phức tạp nhiều hàng cột
* C. Không có sự khác biệt

**Phần 4: Position & Responsive**

**8. position: fixed có đặc điểm?**

* A. Cố định so với viewport
* B. Cố định so với phần tử cha
* C. Không cuộn theo trang

**9. Media query @media (max-width: 768px) áp dụng cho?**

* A. Màn hình > 768px
* B. Màn hình ≤ 768px
* C. Chỉ màn hình = 768px

**Đáp án Mini Test**

**👉 Nhấn vào đây để xem đáp án**

**Phần 1:**

1. **A** - div p (descendant selector)
2. **B** - Chọn phần tử khi rê chuột

**Phần 2:**  
3. **A** - Width bao gồm content + padding + border  
4. **A** - Content → Padding → Border → Margin

**Phần 3:**  
5. **A** - 1 chiều (hàng hoặc cột)  
6. **B** - Chia đều khoảng trống giữa các item  
7. **B** - Layout phức tạp nhiều hàng cột

**Phần 4:**  
8. **A** - Cố định so với viewport  
9. **B** - Màn hình ≤ 768px

**📌 TÓM TẮT CHƯƠNG (Self-Reflection)**

**Những gì tôi đã hiểu:**

1. Hiểu cơ bản về HTML (cấu trúc trang, thẻ cơ bản như div, header, nav, section, article...)
2. Đã nắm được cách liên kết CSS vào HTML dùng một số thuộc tính phổ biến (margin, padding, border, display...).

**Những gì tôi vẫn chưa rõ:**

1. Chưa hiểu cách hoạt động của Flexbox, Grid
2. Phần responsive và media queries vẫn còn hơi mơ hồ khi tự làm mà không có hướng dẫn từng bước.

**Mục tiêu tiếp theo:**

Hiểu hết được nội dung trong slide bài giảng và thuần thục fe để phục vụ cho công việc sau này

**Ghi chú của giáo viên:**

**Ngày hoàn thành:**24/2/2026

**Chữ ký sinh viên:** **Thành**