

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM



HCMUTE

BÁO CÁO CUỐI KÌ: NHÀ THÔNG MINH

Môn học: Thực tập Kiến trúc và giao thức IoT

Mã học phần: ITAL328264_23_1_02

GVHD: Trương Quang Phúc

Ngày, tiết học: Thứ 3 tiết 7 - 11

Nhóm sinh viên:

- 1. Dương Triều Mến - 21139080**
- 2. Cao Tiến Sỹ - 21139051**
- 3. Châu Lê Thủy Tiên - 21139061**

Hồ Chí Minh – 11/202

Thực tập Kiến trúc và Giao thức IoT

A. Đề tài: Nhà thông minh	12
I. Phần cứng.....	12
1. Sơ đồ đặc tả hệ thống	12
2. Sơ đồ khói hệ thống	13
3. Sơ đồ kết nối.....	13
4. Giới thiệu phần cứng.....	13
4.1. ESP32	13
4.2. DHT11 – Cảm biến nhiệt độ và độ ẩm	17
4.3. MQ2 – Cảm biến khí gas	17
4.4. LED đơn	18
II. Phần mềm	19
1. Lưu đồ giải thuật	19
1.1. Lưu đồ giải thuật từ ESP32 tới Firebase	19
1.2. Lưu đồ giải thuật từ Web với Firebase	23
1.3. Lưu đồ giải thuật từ App tới Firebase	25
2. Website	26
2.1. HTML:	26
2.2. CSS:.....	28
2.3 JavaScript:	33
2.4. Chart:.....	39
3. Giao diện điều khiển MIT app	42
III. Trình bày kết quả.....	51
1. Kết quả web HTML	51
1.1. Phòng khách	51
1.2. Phòng ngủ.....	51
1.3. Phòng bếp	52
1.4. Trực quan hóa dữ liệu	52
2. Kết quả phần cứng.....	52

Thực tập Kiến trúc và Giao thức IoT	
3. Kết quả App MIT inventor.....	53
4. Video demo	53
IV. Tài liệu tham khảo.....	53
B. Tìm hiểu HTML, CSS, JS và Google Firebase	54
I. HTML	54
1. HTML Headings	54
2. HTML Paragraphs.....	54
2.1. HTML Paragraphs.....	54
2.2. HTML display:.....	55
2.3. HTML Horizontal Rules:.....	55
2.4. HTML Line Breaks:	56
2.5. The Poem Problem:.....	56
2.6. Giải pháp của ‘Poem problem’: <pre>.....	56
3. HTML Styles.....	57
3.1. Background Color:	57
3.2. Các thuộc tính khác	58
4. HTML Text Formatting	59
5. HTML Quotation and Citation Elements	59
6. HTML Comments	62
7. HTML Colors	63
7.1. Colors	63
7.2. RGB.....	64
7.3. HEX.....	65
7.4. HSL	66
8. HTML CSS	67
8.1. Using CSS	67
8.2. Inline CSS	67
8.3. Internal CSS	68

Thực tập Kiến trúc và Giao thức IoT

8.4. External CSS	68
8.5. CSS Colors, Fonts and Sizes.....	69
8.6. CSS Border.....	70
8.7. CSS Padding.....	70
8.8. CSS Margin	71
8.9. Link to External CSS	71
9. HTML Images	71
9.1. Image	71
9.2. Image Maps	75
9.3. Background Image	77
9.4. The Picture Element	80
10. HTML Tables	82
10.1. HTML Tables.....	82
10.2. HTML Table Border	85
10.3. HTML Table Sizes	92
10.4. HTML Table Headers	95
10.5. HTML Table Padding & Spacing	99
10.6. HTML Table Colspan & Rowspan	102
10.7. Table Styling	104
10.8. Table Colgroup.....	108
11. HTML Lists	115
11.1. List.....	115
11.2. Unordered HTML List	116
11.3. Ordered HTML List	116
11.4. HTML Description Lists	117
12. HTML Block & Inline	117
12.1. Block-level Elements	117
12.2. Inline Elements.....	118

Thực tập Kiến trúc và Giao thức IoT

12.3. The <div> Element.....	119
12.4. The Element	119
13. HTML Iframes	120
13.1. HTML Iframe Syntax.....	120
13.2. Iframe - Set Height and Width.....	120
13.3. Iframe - Remove the Border	121
13.4. Iframe - Target for a Link	121
14. HTML Javascript	122
14.1. The HTML <script> Tag.....	122
14.2. A Taste of JavaScript	123
14.3. The HTML <noscript> Tag.....	123
15. HTML Head	124
15.1. The HTML <title> Element	124
15.2. The HTML <style> Element	124
15.3. The HTML <link> Element	125
15.4. The HTML <meta> Element.....	126
15.5. Setting The Viewport	126
15.6. The HTML <script> Element	126
15.7. The HTML <base> Element	127
16. HTML Layout	127
16.1. HTML Layout Elements	127
16.2. HTML Layout Techniques.....	128
17. HTML Symbols	128
18. HTML Emoji.....	129
19. HTML URL Encode	129
19.1. URL - Uniform Resource Locator	129
19.2. Các URL scheme phổ biến:.....	129
19.3. Mã hóa URL.....	130

Thực tập Kiến trúc và Giao thức IoT

20. HTML Forms	130
20.1. Phần tử <form>	130
20.2. Phần tử <input>	130
20.3. Trường văn bản	130
20.4. Phần tử <label>	131
21. HTML Graphics	132
21.1 HTML Canvas.....	132
21.2 HTML SVG.....	135
22. HTML Video.....	137
22.1. HTML <video> element	137
22.2. HTML <video> autoplay	137
22.3. HTML Video - Methods, Properties, and Events	138
22.4. HTML Video tags	139
23. HTML Audio.....	139
23.1. HTML <audio> autoplay	139
23.2. HTML Audio - Methods, Properties, and Events	140
23.3. HTML Audio tags	140
24. HTML Plug-ins	140
24.1. Plug-ins.....	140
24.2 Phần tử <object>	140
24.3 Phần tử <embed>	141
25. HTML YouTube	142
25.1 Youtube Autoplay + Mute	142
25.2 Youtube Playlists	143
25.3 Youtube Loop.....	143
25.4 Youtube Controls	143
26. HTML Geolocation API	144
II. CSS	146

Thực tập Kiến trúc và Giao thức IoT

1. CSS Colors	146
1.1. Colors	146
1.2. RGB.....	148
1.3. HEX.....	150
1.4. HSL	151
2. CSS Backgrounds.....	152
2.1. Backgrounds color	152
2.2. Background Image	155
2.3. Background Repeat	156
2.4. Background-attachment	158
2.5. Background Shorthand.....	160
3. CSS Border.....	161
3.1. Borders	161
3.2. Border Width.....	162
3.3. Border Color.....	164
3.4. Boder Sides	168
3.5. Border Shorthand	168
3.6. Rounded Borders.....	169
4. CSS Margin	170
4.1. Margin	170
4.2. Margin Collapse	173
5. CSS Padding.....	173
6. CSS Height/Width.....	175
7. CSS Box Model.....	176
8. CSS Outline.....	178
8.1. Outline	178
8.2. Outline width.....	178
8.3. Outline Color.....	179

Thực tập Kiến trúc và Giao thức IoT

8.4. Outline Shorthand	183
8.5. Outline Offset.....	184
9. CSS Text	184
9.1. Text color	184
9.2. Text Alignment	185
9.3. Text Decoration.....	189
9.4 Text Transformation.....	194
9.5 Text Spacing.....	195
9.6 Text Shadow.....	198
10. CSS Fonts	198
10.1 Font Family	198
10.2 Font Web safe.....	199
10.3 Font fallbacks	200
10.4 Fonts style	201
10.5 Font Size.....	204
10.6 Google fonts	207
10.7 Font Pairings	209
10.8 Font shothand	210
11. CSS Icons	211
11.1. Font Awesome Icons.....	211
11.2. Bootstrap Icons.....	212
12. List.....	214
12.1 HTML Lists and CSS List Properties	214
12.2 Các điểm đánh dấu mục danh sách khác nhau	214
12.3 Hình ảnh làm điểm đánh dấu mục danh sách.....	215
12.4 Định vị các điểm đánh dấu mục danh sách	215
12.5 Xóa cài đặt mặc định.....	216
12.6 List - Shorthand property	217

Thực tập Kiến trúc và Giao thức IoT

12.7 Styling list with colors	218
13. CSS Tables	219
13.1 Table borders.....	219
13.2. Table Size.....	221
13.3 Table Alignment.....	222
13.4 Table Style.....	224
13.5 Table Responsive	228
14. CSS Navigation Bar	229
14.1. Vertical Navigation Bar	229
14.2. Horizontal Navigation Bar	234
15. CSS Image Gallery.....	244
16. CSS Website Layout	245
16.1. Website Layout	245
16.2. Header	246
16.3. Navigation Bar	247
16.4. Content	248
16.5. Unequal Columns.....	250
16.6. Footer	253
16.6. Responsive Website Layout.....	255
17. CSS Rounded Corners.....	259
17.1. CSS border-radius Property	259
17.2. CSS border-radius - Specify Each Corner	260
18. CSS Backgrounds.....	261
18.1. CSS Multiple Backgrounds	261
18.2. CSS Background Size	261
18.3. Define Sizes of Multiple Background Images	262
18.4. Full Size Background Image	262
18.5. Hero Image	263

Thực tập Kiến trúc và Giao thức IoT

18.6. CSS background-origin Property	264
18.7. CSS background-clip Property.....	265
19. CSS Buttons	266
19.1. Basic Button Styling	266
19.2. Button Colors	266
19.3. Button Sizes	267
19.4. Rounded Buttons	269
19.5. Colored Button Borders	270
19.6. Hoverable Buttons.....	272
19.7. Shadow Buttons	274
19.8. Disabled Buttons	275
19.9. Button Width	276
19.10. Button Groups	277
19.11. Bordered Button Group.....	278
19.12. Vertical Button Group.....	279
19.13. Button on Image	280
19.14. Animated Buttons.....	281
20. CSS Object- position.....	282
20.1. The Image.....	282
20.2. Using the object-position Property.....	283
21. CSS Multiple Columns	284
21.1. CSS Multi-column Layout	284
21.2. CSS Create Multiple Columns	284
21.3. CSS Specify the Gap Between Columns	285
21.4. CSS Column Rules.....	286
21.5. Specify How Many Columns an Element Should Span	290
21.6. Specify The Column Width	291
III. Google Firebase.....	292

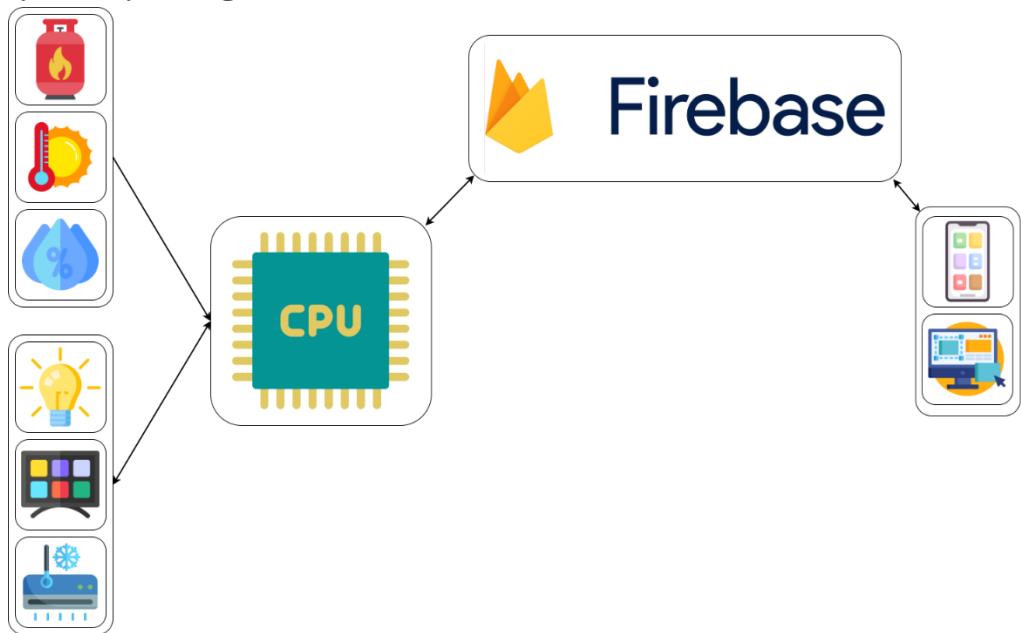
Thực tập Kiến trúc và Giao thức IoT

1. Giới thiệu.....	292
2. Ưu nhược điểm.....	292
2.1. Ưu điểm:.....	292
2.2. Nhược điểm:	292
3. Code	293
3.1 HMTL.....	293
3.2 CSS	294
3.3 Javascrip.....	295
THÀNH VIÊN NHÓM	296

A. Đề tài: Nhà thông minh

I. Phần cứng

1. Sơ đồ đặc tả hệ thống

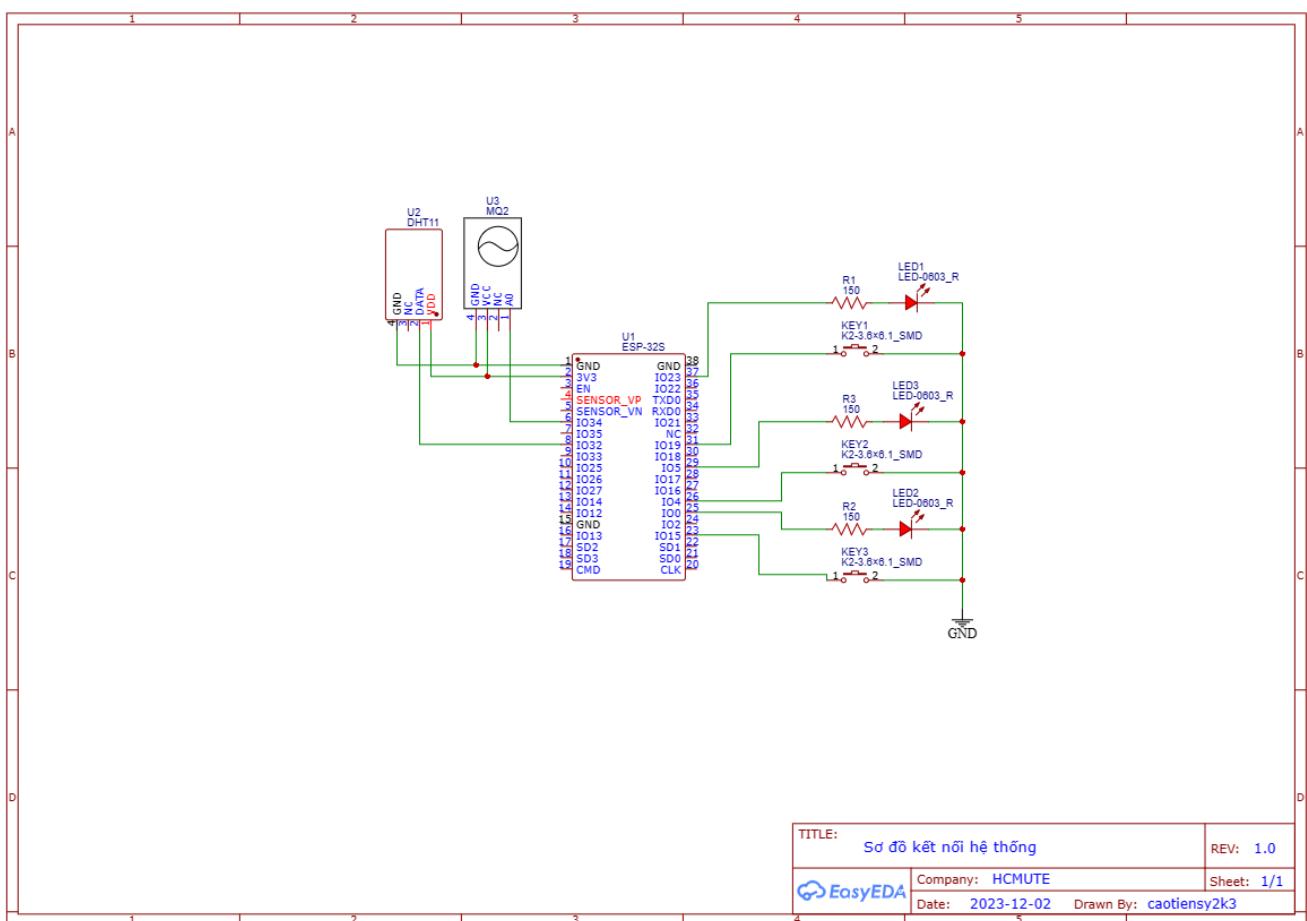


Hình 1: Sơ đồ hệ thống nhà thông minh

Thực tập Kiến trúc và Giao thức IoT

2. Sơ đồ khối hệ thống

3. Sơ đồ kết nối

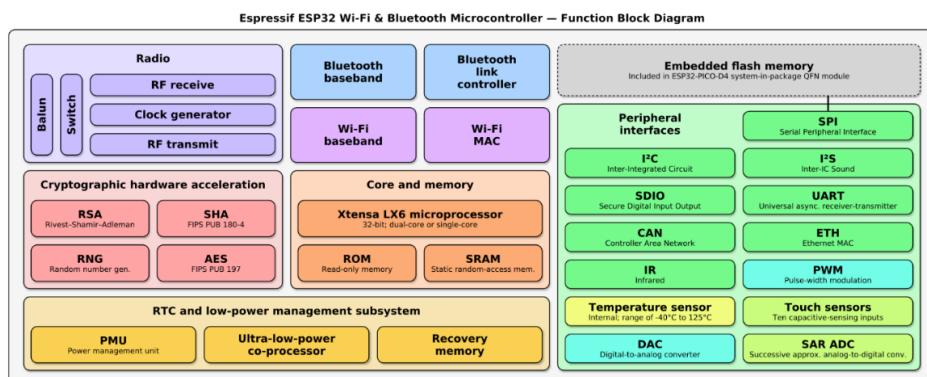


Hình 2: Sơ đồ nối chân các cảm biến và thiết bị

4. Giới thiệu phần cứng

4.1. *ESP32*

- ESP32 là vi điều khiển giá rẻ với công suất thấp được tích hợp Wifi và Bluetooth kép (dual Bluetooth). ESP32 bao gồm các thành phần như nhân xử lý trung tâm, các công tắc ăng ten tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại tín hiệu low-noise (nhiều thấp), bộ lọc và mô-đun quản lý nguồn. ESP32 được sản xuất bởi TSMC với quy trình 40nm.



Hình 3: Sơ đồ khối ESP32

Thực tập Kiến trúc và Giao thức IoT

- Giới thiệu chung về ESP32-S3:

- + Vi điều khiển ESP32 sử dụng CPU Dual-core Xtensa LX7, có tốc độ lên đến 240 MHz, và hỗ trợ FPU (Floating Point Unit) đơn chính xác.
- + 512 KiB SRAM, 384 KiB ROM, and 16 KiB RTC SRAM.
- + Có khả năng kết nối với PSRAM và Flash bên ngoài thông qua giao diện Quad SPI hoặc Octal SPI, và chia sẻ cùng không gian địa chỉ 32 MiB.
- + Bộ xử lý RISC-V siêu tiết kiệm năng lượng (RV32IMC), hoạt động ở tần số khoảng 17.5 MHz.
- + Bộ xử lý FSM (Finite State Machine) siêu tiết kiệm năng lượng, tương tự như các phiên bản trước của ESP32 và ESP32-S2.
- + Wi-Fi 2.4 GHz (IEEE 802.11 b/g/n) cho kết nối không dây.
- + Bluetooth 5 (LE) để hỗ trợ kết nối Bluetooth Low Energy.
- + 44 chân GPIO có thể được lập trình để thực hiện các chức năng khác nhau.
- + Không tích hợp Ethernet MAC (Media Access Control) tích hợp.
- + 2 × 12-bit SAR ADCs (Analog-to-Digital Converters) với khả năng xử lý tới 20 kênh đầu vào.
- + USB OTG (On-The-Go) cho khả năng kết nối USB linh hoạt.

* Đặc tính

- *Bluetooth:*

- + Tuân thủ các đặc tả Bluetooth v4.2 BR/EDR và Bluetooth LE.
- + Truyền tải với Class-1, Class-2 và Class-3 mà không cần bộ khuếch đại công suất bên ngoài.
- + Enhanced Power Control.
- + Công suất truyền tải +9 dBm.
- + Bộ thu NZIF với độ nhạy Bluetooth LE là -94 dBm.
- + Adaptive Frequency Hopping (AFH).
- + Giao diện HCI chuẩn dựa trên SDIO/SPI/UART.
- + HCI UART tốc độ cao, lên đến 4 Mbps.
- + Điều khiển đa chế độ Bluetooth 4.2 BR/EDR và Bluetooth LE.
- + Kết nối đồng bộ dựa trên SCO/eSCO.
- + CVSD và SBC cho codec âm thanh.
- + Bluetooth Piconet và Scatternet.
- + Đồng thời quảng cáo và quét.

- *Wifi:*

- + Chuẩn 802.11b/g/n.
- + Tốc độ lên đến 150 Mbps ở 802.11n (2.4 GHz).
- + Hỗ trợ WMM (Wi-Fi Multimedia).
- + TX/RX A-MPDU, RX A-MSDU (Aggregated MAC Protocol Data Unit).

Thực tập Kiến trúc và Giao thức IoT

- + Chức năng Immediate Block ACK.
- + Chức năng Defragmentation.
- + Beacon monitoring tự động (theo đồng hồ thời gian thực phần cứng).
- + 4 giao diện Wi-Fi ảo.
- + Hỗ trợ đồng thời các chế độ Infrastructure Station, SoftAP và Promiscuous.
- + Đa dạng anten.

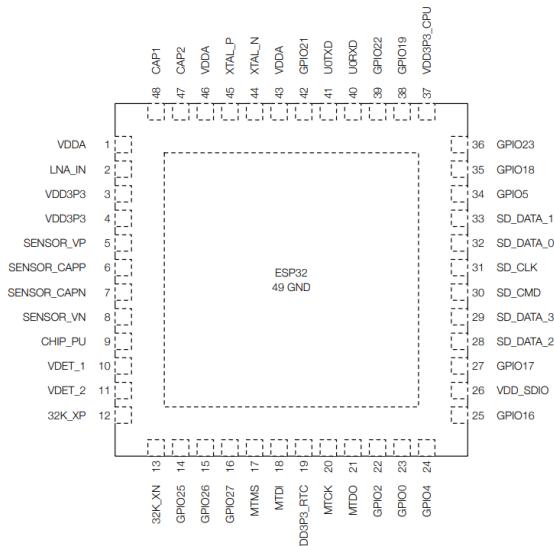
- **CPU và bộ nhớ:**

- + Bộ xử lý Xtensa 32-bit LX6 đơn hoặc đôi.
- + Điểm số CoreMark:
 - 1 nhân ở 240 MHz: 504.85 CoreMark; 2.10 CoreMark/MHz.
 - 2 nhân ở 240 MHz: 994.26 CoreMark; 4.14 CoreMark/MHz.
- + ROM 448 KB.
- + SRAM 520 KB.
- + SRAM trong RTC 16 KB.
- + QSPI hỗ trợ nhiều chip flash/SRAM.

- **Clock và timer:**

- + Dao động nội 8 MHz với hiệu chuẩn.
- + Dao động RC nội với hiệu chuẩn.
- + Dao động tinh thể ngoại 2 MHz ~ 60 MHz (40 MHz chỉ cho chức năng WiFi/Bluetooth).
- + Dao động tinh thể ngoại 32 kHz cho RTC với hiệu chuẩn.
- + Hai nhóm bộ hẹn giờ, bao gồm 2 × bộ hẹn giờ 64-bit và 1 × bộ hẹn giờ watchdog chính trong mỗi nhóm.
 - + Một bộ hẹn giờ RTC.
 - + Đồng hồ bảo vệ RTC.

- **GPIO:**



Hình 4: Sơ đồ chân kết nối ESP32

GVHD: Trương Quang Phúc

Thực tập Kiến trúc và Giao thức IoT

- + 34 GPIO có thể lập trình.
- + 5 GPIO dùng cho cài đặt.
- + 6 GPIO chỉ đầu vào.
- + 6 GPIO cần thiết cho flash/PSRAM trong gói (ESP32-D0WDR2-V3, ESP32-U4WDH).
- + ADC SAR 12-bit lên đến 18 kênh.
- + 2 DAC 8-bit.
- + 10 cảm biến chạm.
- + 4 SPI.
- + 2 I2S.
- + 2 I2C.
- + 3 UART.
- + 1 host (SD/eMMC/SDIO).
- + 1 slave (SDIO/SPI).
- + Giao diện Ethernet MAC với DMA và hỗ trợ IEEE 1588.
- + TWAI®, tương thích với ISO 11898-1 (CAN Specification 2.0).
- + RMT (TX/RX).
- + Motor PWM.
- + LED PWM lên đến 16 kênh.

- Quản lý nguồn:

- + Kiểm soát năng lượng chi tiết thông qua lựa chọn tần số đồng hồ, tỷ lệ nhiệm vụ, các chế độ hoạt động Wi-Fi và kiểm soát năng lượng cá nhân của các thành phần nội bộ.
- + Năm chế độ nguồn dành cho các tình huống điển hình: Active, Modem-sleep, Light-sleep, Deep-sleep, Hibernation.
- + Tiêu thụ năng lượng ở chế độ Deep-sleep là 10 μ A.
- + Bộ xử lý siêu tiết kiệm năng lượng (ULP).
- + Bộ nhớ RTC vẫn được cấp nguồn trong chế độ Deep-sleep.

- Bảo mật:

- + Khởi động an toàn (Secure boot).
- + Mã hóa flash.
- + OTP 1024-bit, tối đa 768-bit cho khách hàng.
- + Tăng tốc phần cứng mã hóa: AES, Hash (SHA-2), RSA, ECC, Random Number Generator (RNG).

- **Các lĩnh vực ứng dụng:** Với việc tiêu thụ năng lượng thấp, ESP32 là lựa chọn lý tưởng cho các thiết bị IoT trong các lĩnh vực sau đây:

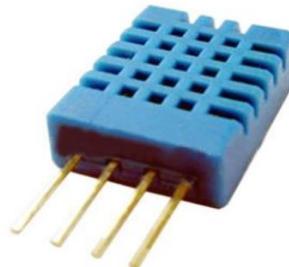
- + Nhà Thông Minh
- + Tự Động Hóa Công Nghiệp
- + Chăm Sóc Sức Khỏe

Thực tập Kiến trúc và Giao thức IoT

- + Điện Tử Tiêu Dùng
- + Nông Nghiệp Thông Minh
- + Máy POS
- + Robot Dịch Vụ
- + Thiết Bị Âm Thanh
- + Trung Tâm Cảm Biến IoT Tiêu Thu Năng Lượng Thấp
- + Thiết Bị Ghi Dữ Liệu IoT Tiêu Thu Năng Lượng Thấp
- + Camera Cho Phát Video
- + Nhận Dạng Giọng Nói
- + Nhận Dạng Hình Ảnh
- + Thẻ Mạng SDIO Wi-Fi + Bluetooth
- + Cảm Ứng và Nhận Biết Gần

4.2. DHT11 – Cảm biến nhiệt độ và độ ẩm

- Cảm biến Nhiệt Độ và Độ Ẩm DHT11 có tính năng là một cảm biến nhiệt độ và độ ẩm kết hợp với đầu ra tín hiệu số được hiệu chỉnh. Đảm bảo độ tin cậy cao và ổn định lâu dài. Cảm biến này bao gồm một thành phần đo độ ẩm kiểu điện trở và một thành phần đo nhiệt độ NTC.



Hình 5: Cảm biến DHT11

* Các thông số kỹ thuật:

	Nhiệt độ	Độ ẩm
Khoảng đo được	-0 đến 50°C.	+ Điều kiện 0°C: 30% tới 90% + Điều kiện 25°C: 20% tới 90% + Điều kiện 50°C: 20% tới 80%
Thời gian phản hồi	- từ 6 đến 30 giây.	- Ở điều kiện chuẩn: trung bình 10s giây (tối thiểu: 6 giây – tối đa: 15 giây).
Sai số	- từ 1 đến 2%	- Sai số trung bình ở 25°C: 4%

Bảng 1: Thông số kỹ thuật DHT11

4.3. MQ2 – Cảm biến khí gas

- MQ2 là cảm biến khí, dùng để phát hiện các khí có thể gây cháy. Được cấu tạo từ bán dẫn SnO₂. Chất này có độ nhạy cảm thấp với không khí sạch và nhạy cảm cao với GVHD: Trương Quang Phúc

Thực tập Kiến trúc và Giao thức IoT

môi trường có khí gây cháy. Nhờ tính chất này người ta thêm vào mạch đơn giản để biến đổi độ nhạy thành điện áp.

- Khi môi trường sạch, điện áp đầu ra của cảm biến thấp. Giá trị điện áp càng tăng khi lượng khí gây cháy càng cao.

- MQ2 hoạt động rất tốt trong môi trường khí hóa lỏng LPG, H₂, và các chất khí gây cháy khác. Nó được sử dụng rộng rãi trong công nghiệp bởi mạch đơn giản và chi phí thấp.



Hình 6: Sơ đồ chân cảm biến MQ-2

- + VCC: Chân nối nguồn
- + GND: Chân nối đất
- + DO (Digital Out): Chân này để nhận đầu ra kỹ thuật số bằng cách đặt một giá trị ngưỡng sử dụng potentiometer (bộ điều chỉnh điện trở).
- + AO (Analog Out): Chân này đầu ra điện áp tương tự từ 0.3 đến 4.5V dựa trên cường độ của khí.

4.4. LED đơn

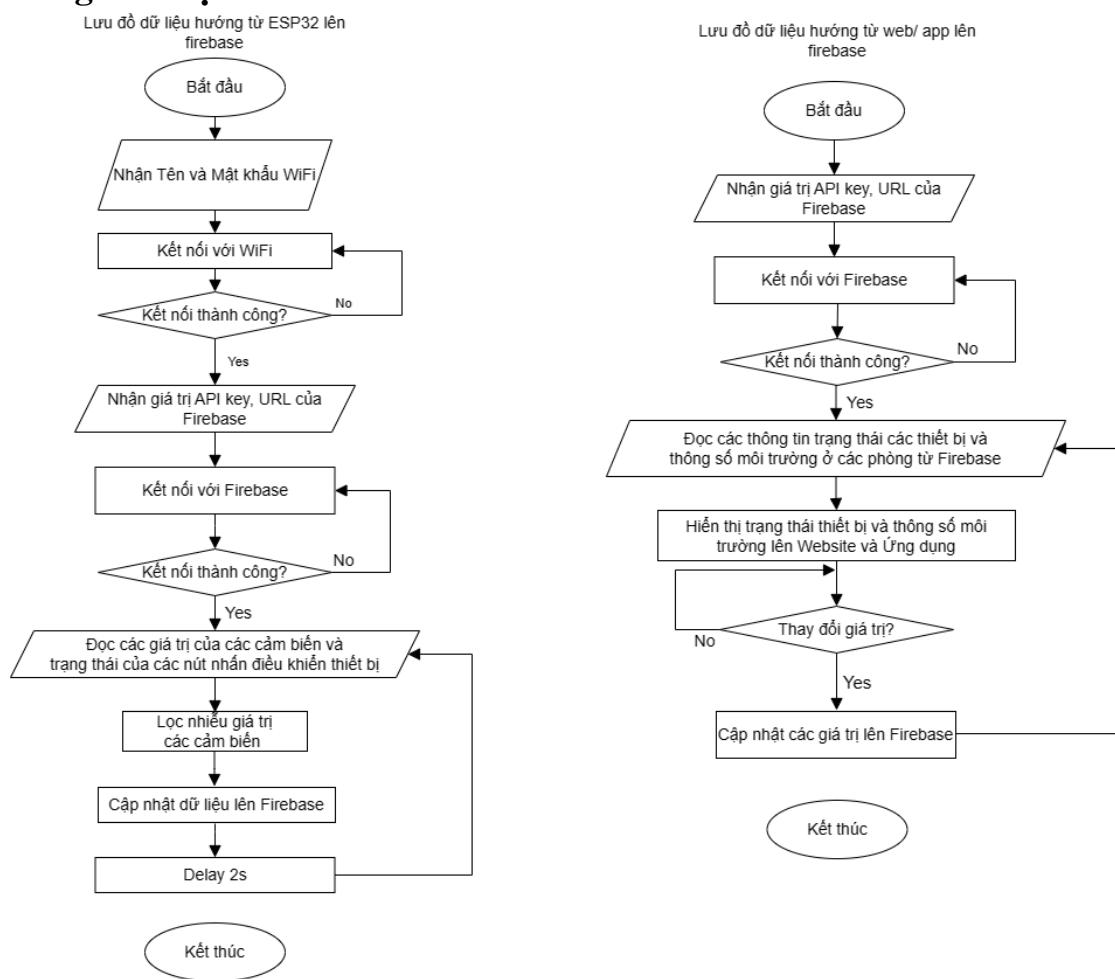
- Mỗi Led đơn đại diện cho 1 thiết bị trong nhà để thể hiện chức năng bật tắt thiết bị mà ứng dụng có được.

- Thông số kỹ thuật:
 - + Điện áp: 1.9 – 2.1 V
 - + Dòng: 10 – 20 mA
 - + Độ sáng: 5000 – 7000 MCD
 - + Loại led: led phủ
 - + Kích thước: 3mm

Thực tập Kiến trúc và Giao thức IoT

II. Phần mềm

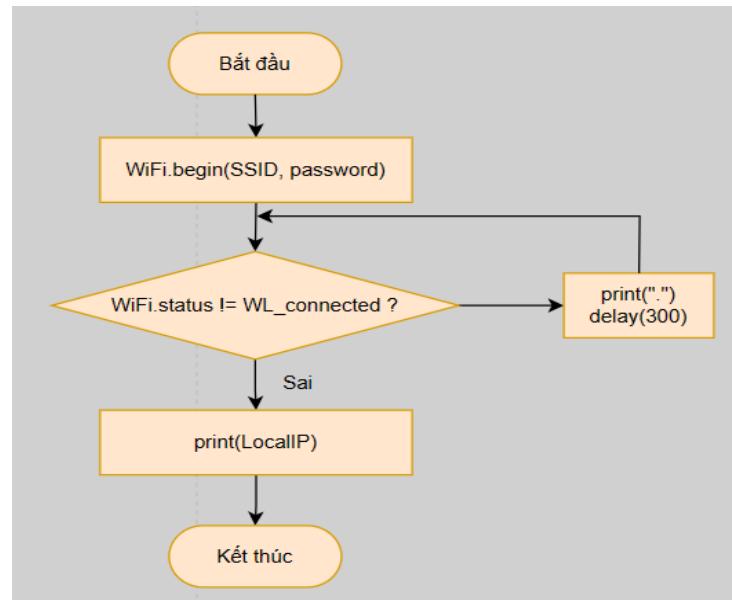
1. Lưu đồ giải thuật



Hình 7: Lưu đồ giải thuật của hệ thống

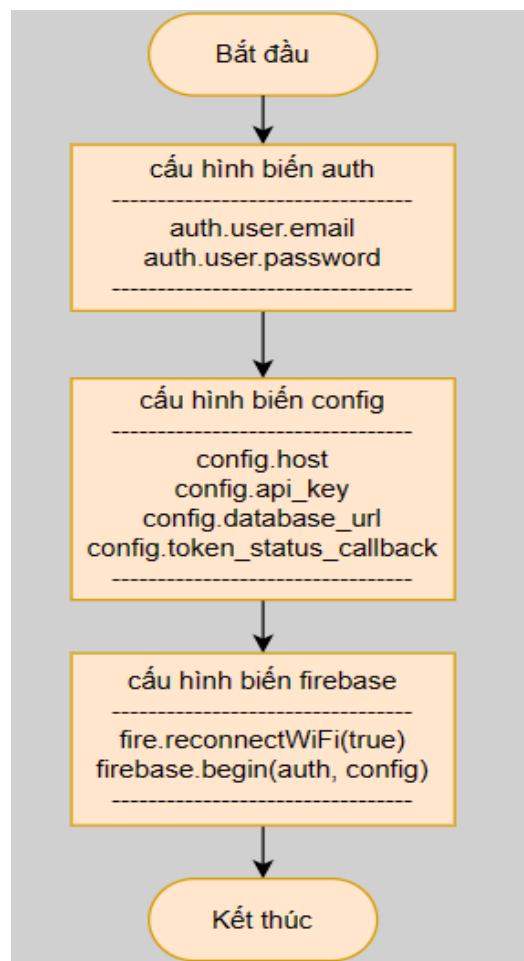
1.1. Lưu đồ giải thuật từ ESP32 tới Firebase

a, Kết nối wifi

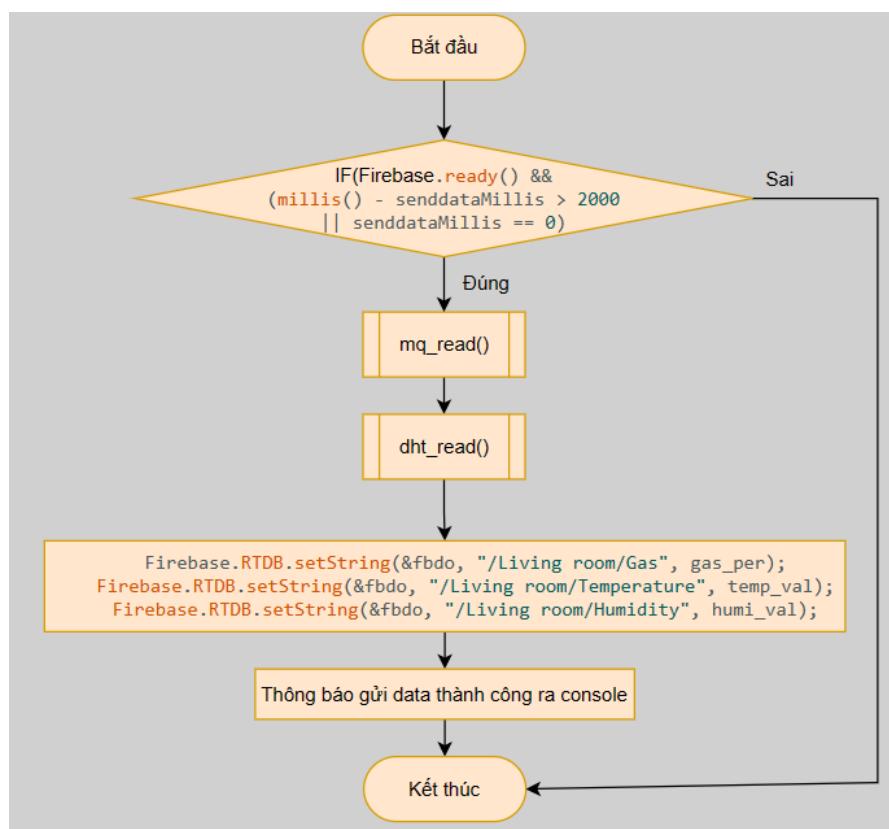


Thực tập Kiến trúc và Giao thức IoT

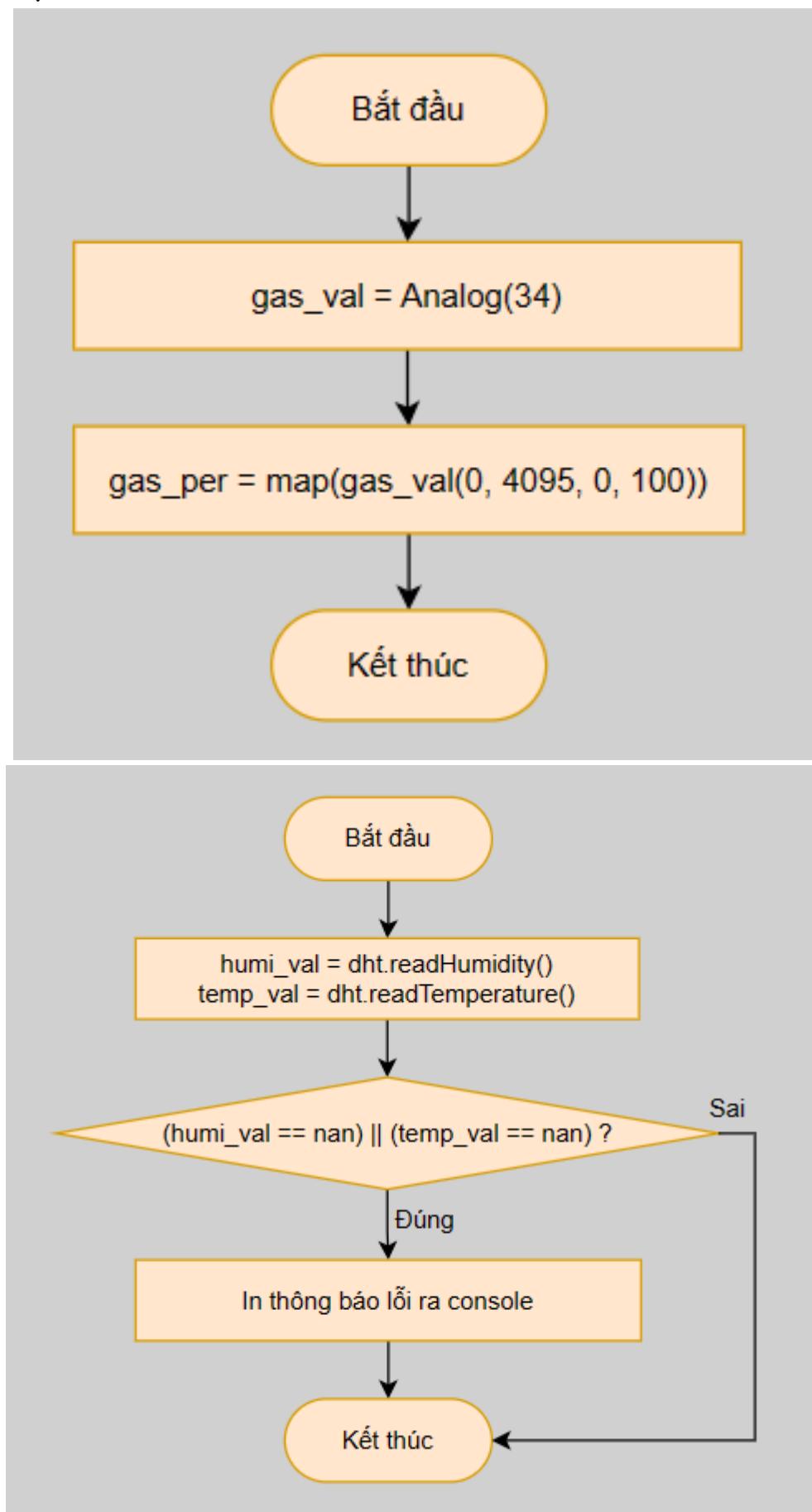
b, Kết nối với Firebase



c, Gửi dữ liệu lên Firebase

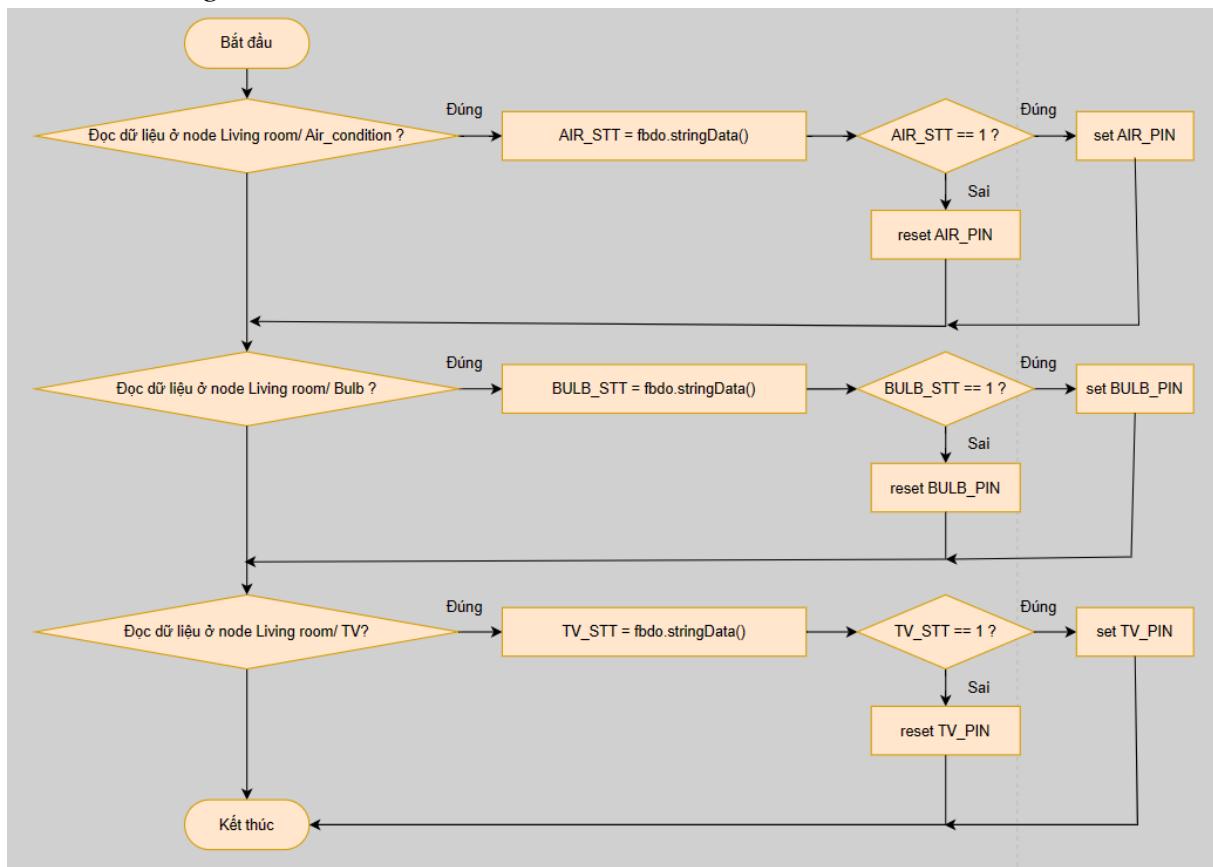


d, Đọc dữ liệu cảm biến

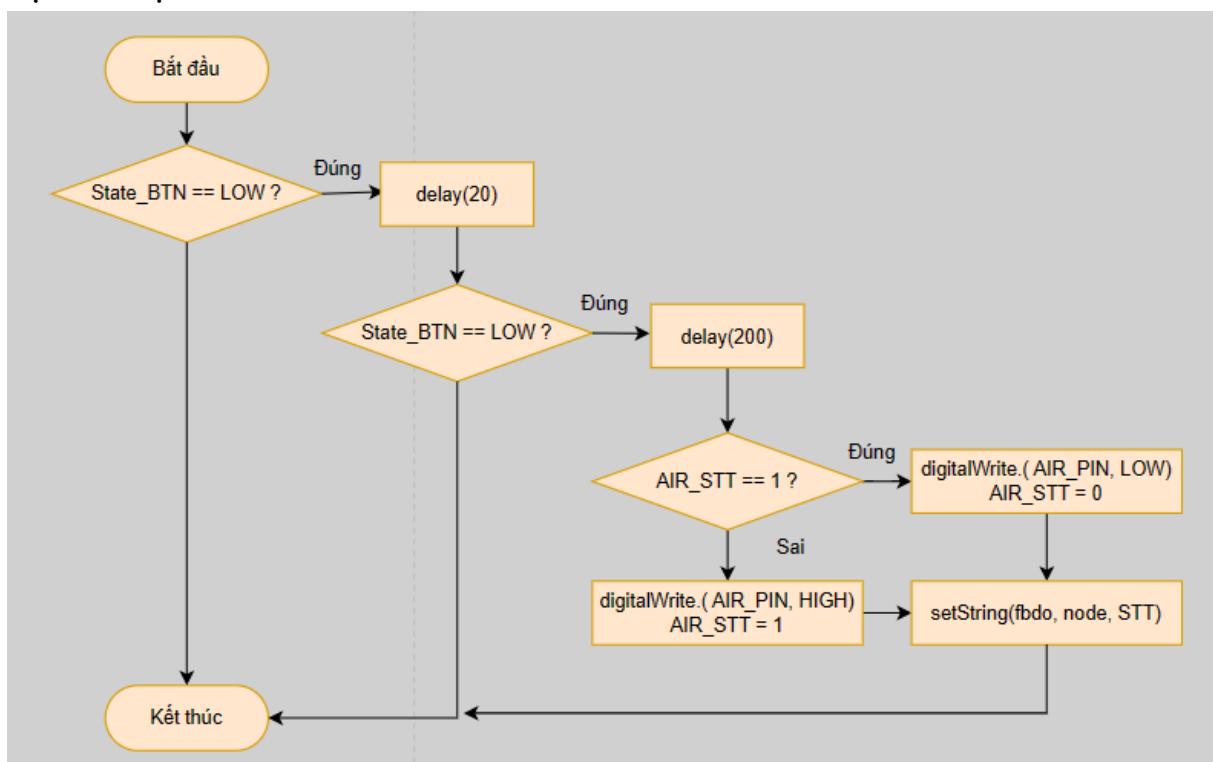


Thực tập Kiến trúc và Giao thức IoT

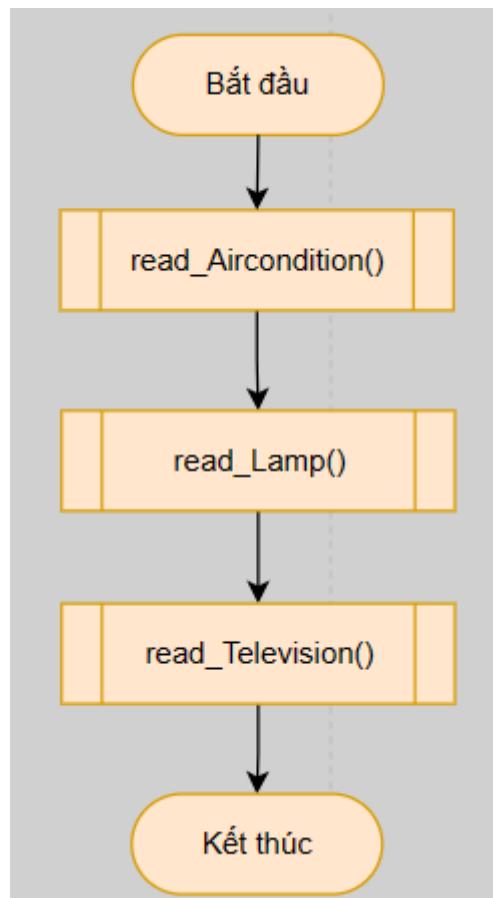
e, Hiển thị trạng thái của các thiết bị



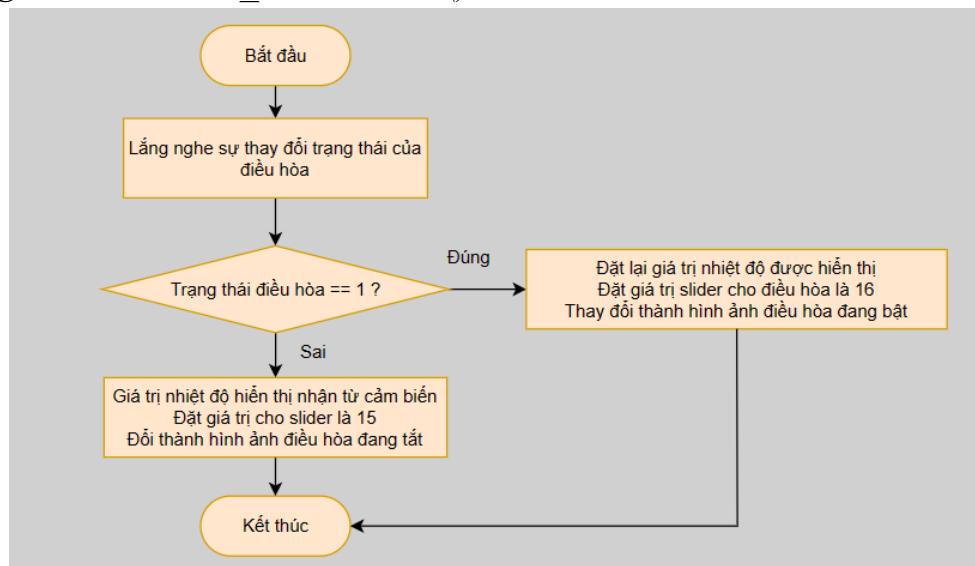
f, Đọc tín hiệu nút nhấn



1.2. Lưu đồ giải thuật từ Web với Firebase

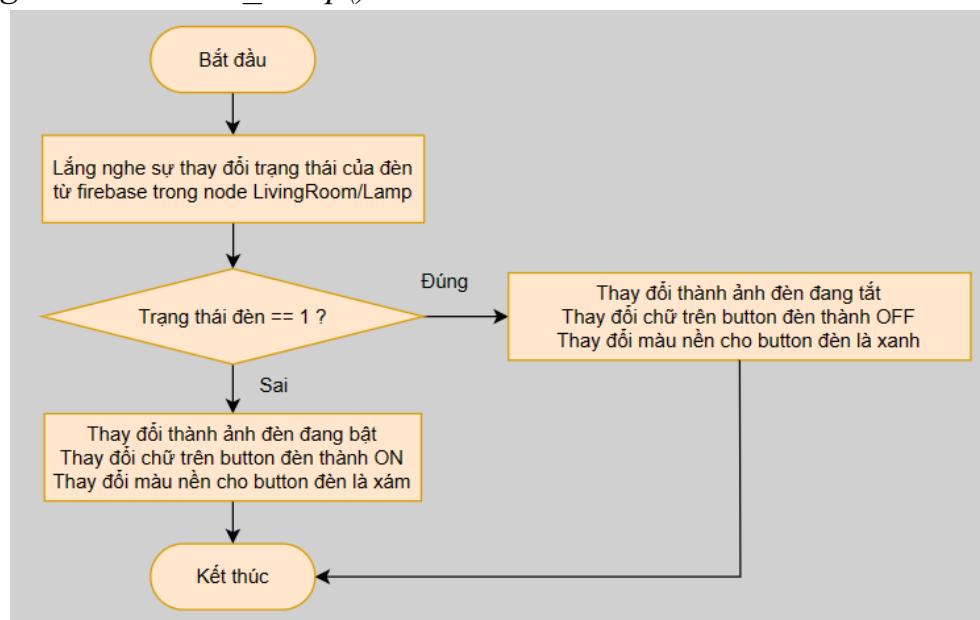


a, Chương trình con read_Aircondition()

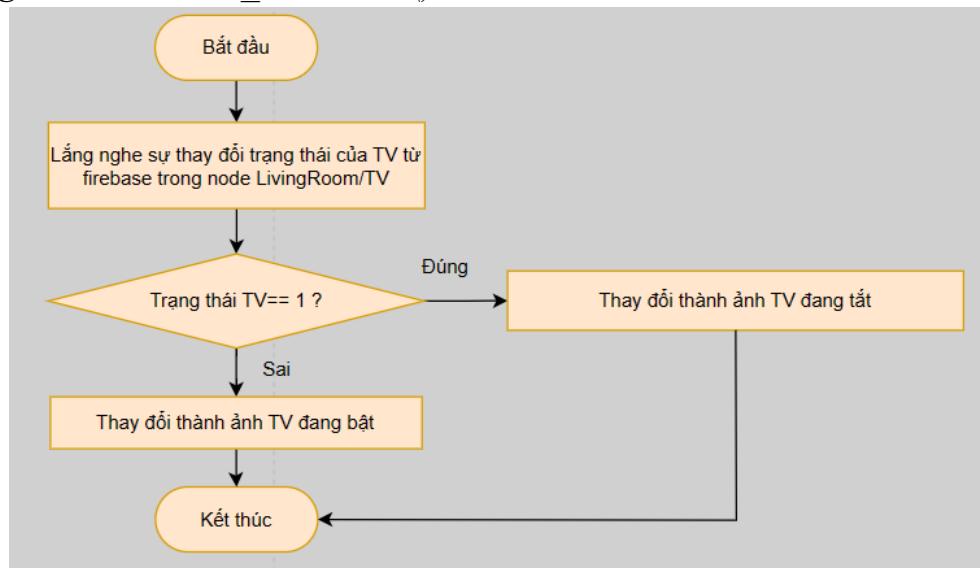


Thực tập Kiến trúc và Giao thức IoT

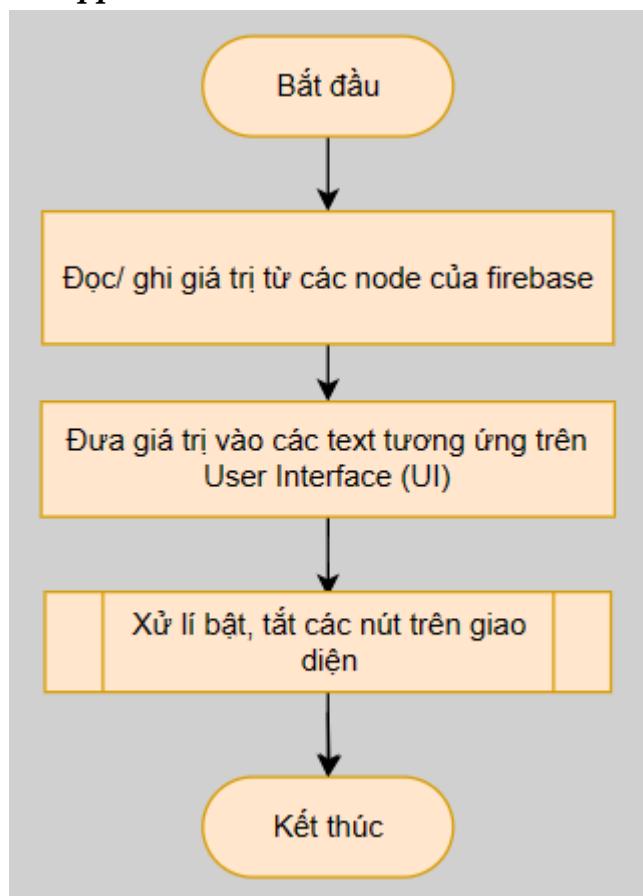
b, Chương trình con read_Lamp()



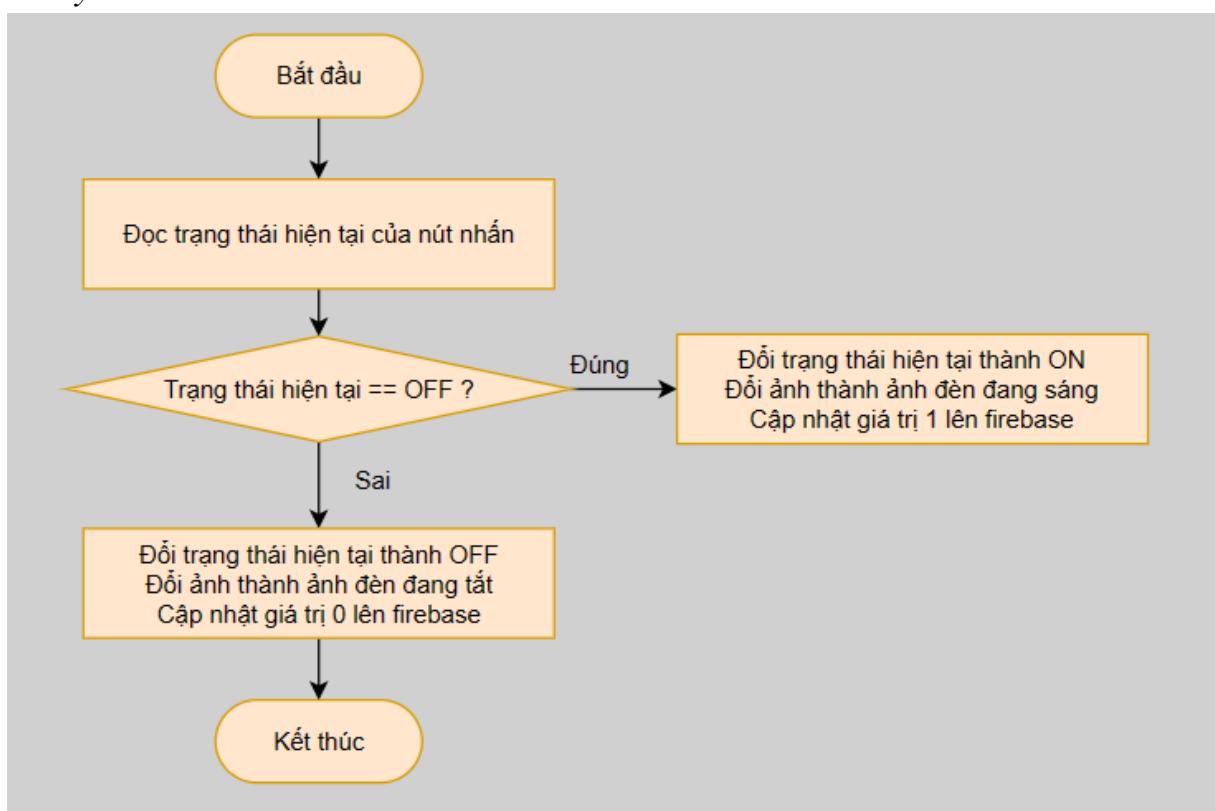
c, Chương trình con read_Television()



1.3. Lưu đồ giải thuật từ App tới Firebase

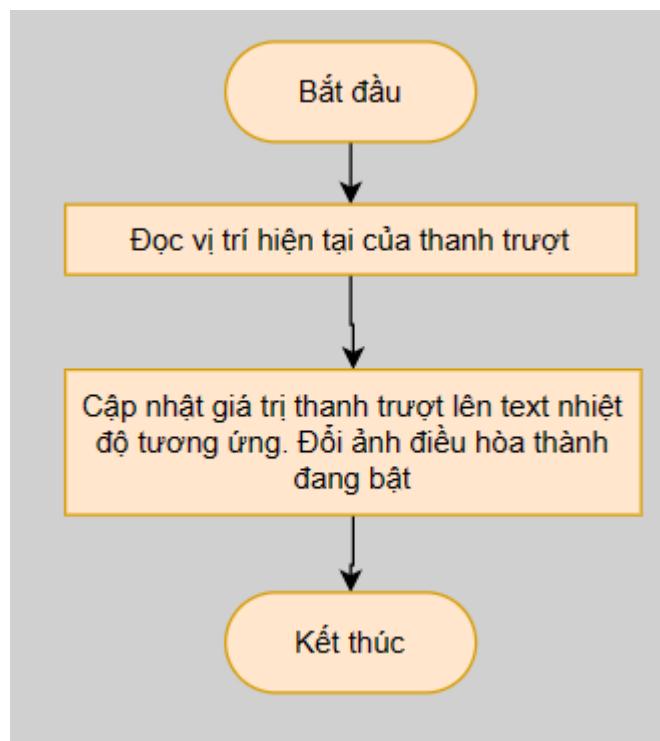


a, Xử lý cho nút nhấn ON/OFF

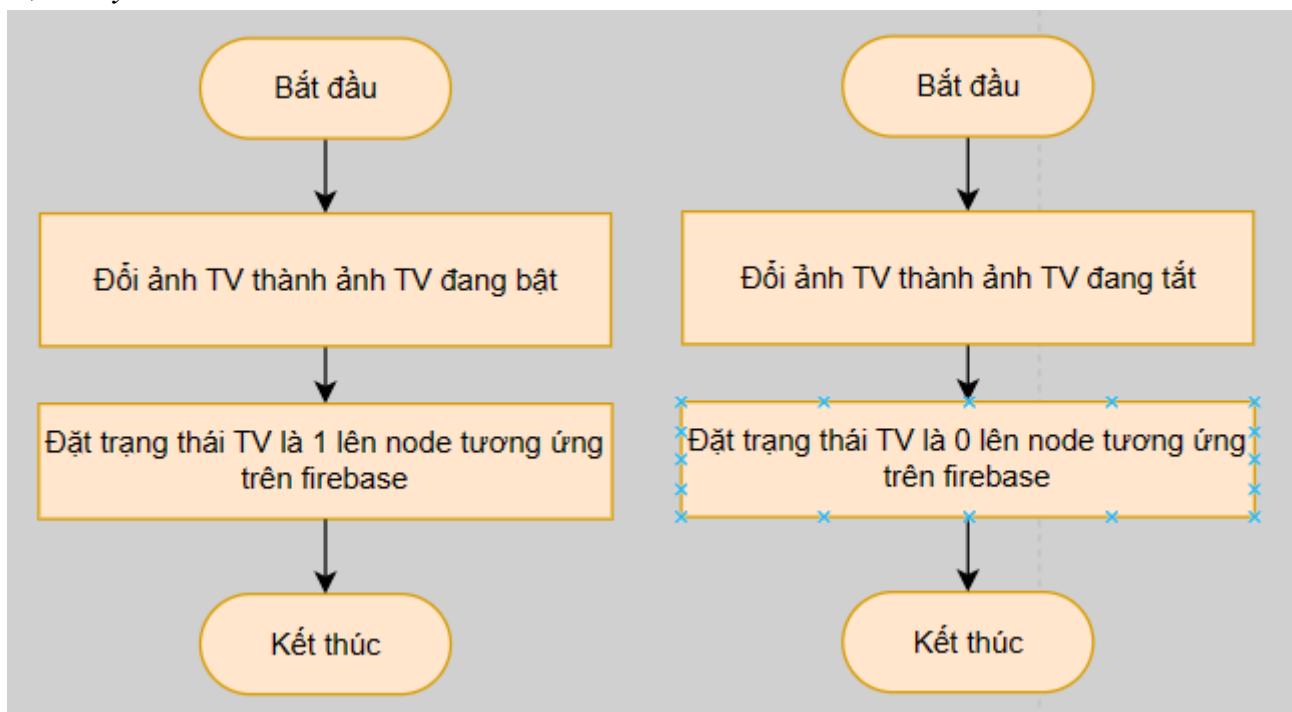


Thực tập Kiến trúc và Giao thức IoT

b, Xử lý thanh trượt



c, Xử lý cho 2 nút ON và OFF



2. Website

2.1. HTML:

- **Thẻ “head” và “meta”:** Trang web được bắt đầu với khai báo HTML cơ bản bao gồm thẻ tiêu đề và các thẻ meta để đảm bảo chuẩn mực hóa và xác định cách trình duyệt hiển thị trang.

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>SMART HOME</title>
    <meta name="description" content="">
```

- **Định dạng trang:** Để đảm bảo trang web hiển thị tốt trên các thiết bị khác nhau, đã sử dụng thẻ meta để điều chỉnh tỷ lệ hiển thị và liên kết tới các file CSS và font từ Google Font API.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="style.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=AudioWide|Sofia|Trirong">
```

- **Header và Menu Tab:** Phần header chứa tiêu đề của trang web và một phần hiển thị thời gian. Các tab điều hướng giữa các phòng trong Smart Home được đặt ở phần topnav.

```
<div class="header">
  <h1>SMART HOME</h1>
  <p id="time"></p>
</div>

<div class="topnav">
  <button class="tablinks" onclick = "ClickLivingRoom()">Living Room</button>
  <button class="tablinks" onclick = "ClickBedRoom()">Bedroom</button>
  <button class="tablinks" onclick = "ClickKitchen()">Kitchen</button>
  <button class="tablinks" onclick = "ClickVD()">Visualize Data</button>
</div>
```

- **Phần hiển thị thông tin của các phòng:** Mỗi phòng (Living Room, Bedroom, Kitchen) được hiển thị bằng cách sử dụng các div với class "room". Mỗi phòng có các thông tin cụ thể như nhiệt độ, độ ẩm, khí gas và các thiết bị như điều hòa, đèn, TV... Cụ thể:

+ Thẻ '.row': Đây là một div tổng quát chứa tất cả các phòng và nội dung liên quan đến giao diện điều khiển. Thẻ này dùng để tạo cấu trúc layout chính của trang web.

+ Các div '.room': Mỗi div này đại diện cho một phòng cụ thể trong ngôi nhà. Dùng để chứa thông tin và điều khiển các thiết bị trong từng phòng.

+ Thẻ '.card': Đây là một khung chứa thông tin về các yếu tố trong phòng. Dùng để phân chia và nhóm các thông tin và thiết bị trong phòng.

+ Các div '.column' : Các div này nằm bên trong .card và chứa thông tin về nhiệt độ, độ ẩm, khí gas và các thiết bị trong phòng. Dùng để hiển thị thông tin cụ thể và các nút điều khiển cho từng thiết bị.

+ Các button: Các nút này đi kèm với các thiết bị như điều hòa, đèn, TV... Có chức năng điều khiển trạng thái bật/tắt của các thiết bị trong phòng.

+ Phần biểu đồ '.chart-container': Đây là một phần riêng biệt chứa biểu đồ cho mỗi phòng. Có chức năng để hiển thị dữ liệu số liệu trong biểu đồ cho người dùng theo dõi.

- **Phần Footer:** Footer chứa thông tin về trường đại học, khoa, môn học và các thông tin liên quan đến bài báo cáo thực tập. Ngoài ra, còn có thông tin về nội dung và thành viên trong nhóm làm việc.

Thực tập Kiến trúc và Giao thức IoT

```
<div class="footer">
    <div class = column-footer>
        <p>Trường Đại học Sư phạm Kỹ Thuật - Hồ Chí Minh University Technology Education<br>
            Khoa: Điện - Điện tử <br>
            Môn học: Thực tập Kiến trúc và Giao thức IoT<br>
            Lớp: 21139 <br>
            Bài báo cáo thực tập <br>
    </p>
    </div>
    <div class = column-footer>
        <p> Nội dung: Thiết kế giao diện người dùng cho ứng dụng nhà thông minh<br>
            Nhóm sinh viên: <br>
            1. Cao Tiến Sỹ <br>
            2. Châu Lê Thúy Tiên <br>
            3. Dương Triều Mến <br>
    </p>
    </div>
</div>
```

- **Phần script và liên kết tới Firebase:** Trang web kết nối tới Firebase để đồng bộ và lưu trữ dữ liệu. Sử dụng các script từ Chart.js để hiển thị dữ liệu dưới dạng biểu đồ và các script khác để điều khiển thiết bị từ xa.

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels@2.0.0"></script>

<script src="https://www.gstatic.com/firebasejs/8.2.10/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio-analytics.js"></script>
<script src=".main.js"></script>
```

2.2. CSS:

- Header:

+ ‘.header’: Định dạng khung chứa tiêu đề của trang web, có background màu, canh giữa, padding và font-size cụ thể.

+ ‘.header h1’: Định dạng phần tiêu đề chính với font-size khác.

```
.header {
    display: block;
    width: 100%;
    height: auto;
    background-color: hwb(178 14% 16%);
    text-align: center;
    font-size: 40px;
    font-family: "Audiorwide", sans-serif;
    padding: 10px;
}

.header h1
{
    font-size: 40px;
}
```

Thực tập Kiến trúc và Giao thức IoT

- Top Navigation Bar:

+ '.topnav': Tạo khung chứa thanh điều hướng, với màu nền và overflow ẩn nội dung bị tràn.

+ '.topnav button': Định dạng các nút trong thanh điều hướng, bao gồm cả hover và trạng thái active.

```
/* Style the top navigation bar */
.topnav {
    overflow: hidden;
    background-color: #333;
}

/* Style the topnav links */
.topnav button {
    float: left;
    display: block;
    color: #f2f2f2;
    background-color: #333;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    border: none;
    width: 25%;
    font-family: 'Audiorwide', sans-serif;
    font-size: 24px;
    font-weight: bold;
}

/* Change color on hover */
.topnav button:hover {
    background-color: #696464;
    color: rgb(15, 14, 14);
}
.topnav button.active {
    font-weight: bold;
    background-color: #ddd;
    color: rgb(8, 8, 8);
}
```

- Card: Định dạng các khối card trên trang web. Đây là các phần hiển thị thông tin như thông tin phòng, thiết bị, v.v.

```
.card {
    background-color: white;
    padding: 0px;
    margin: 0px;
}
```

- Row: Tạo các dòng chứa các cột, các phần tử trong trang có thể sắp xếp theo dòng này.

```
.row{  
    position: relative;  
}  
/* Clear floats after the columns */  
.row::after {  
    content: "";  
    display: table;  
    clear: both;  
}
```

- Column: Định dạng các cột, là các phần tử chứa thông tin cụ thể như nhiệt độ, độ ẩm, v.v. Mỗi loại thông tin có thể có màu sắc và kích thước chữ khác nhau.

```
.column  
{  
    float: left;  
    width: 33.333333%;  
    text-align: center;  
    font-family: 'Audiorwide', sans-serif;  
    font-size: 24px;  
    font-weight: bold;  
    color: #0c0c0c;  
}
```

- Image: Định dạng kích thước tối đa của hình ảnh trong các phần tử trên trang.

```
.image  
{  
    max-width: 40%;  
    max-height: 40%;  
}
```

- bgr_img: Tạo một hình nền mờ, làm nổi bật, tạo hiệu ứng đẹp mắt cho phần nội dung của trang web. Mỗi phòng sẽ có hình nền là phòng tương ứng.

Thực tập Kiến trúc và Giao thức IoT

```
.bgr_img{  
    position: absolute;  
    padding: auto;  
    width: 100%;  
    height: 100%;  
    background-image: url("img/living-room-background.jpg");  
    background-repeat: no-repeat;  
    filter: blur(3px); /* Điều chỉnh độ mờ tại đây */  
    z-index: -1;  
    background-position: center;  
    background-size: cover;  
}
```

- footer: Định dạng và sắp xếp vị trí của phần footer, chứa thông tin về nhóm thực hiện báo cáo.

```
.footer  
{  
/*     position: fixed; */  
    left: 0;  
    bottom: 0;  
    width: 100%;  
    background-color: #f9f9f9;  
    padding: 0;  
}  
  
.column-footer{  
    font-size: 15px;  
    float: left;  
    width: 50%;  
    font-weight: bold;  
    background-color: rgba(0, 0, 0, 0.764);  
    color: rgba(255, 255, 255, 0.588);  
    padding: 0px 60px;  
}
```

Thực tập Kiến trúc và Giao thức IoT

- btn: Định dạng các button (các nút bật/tắt các thiết bị), bao gồm khi hover, active.

```
.btn {  
    background-color: #4CAF50; /* Màu nền khi bật */  
    color: white; /* Màu chữ khi bật */  
    border: none;  
    padding: 10px 20px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
    font-size: 16px;  
    font-weight: bold;  
    cursor: pointer;  
    border-radius: 5px;  
    margin: 5px;  
    transition: background-color 0.3s, color 0.3s; /* Hiệu ứng hover */  
}  
  
.btn:hover {  
    background-color: #324d33; /* Màu nền khi hover */  
}  
.btn.active {  
    background-color: #223623; /* Màu nền khi nút được nhấn */  
    color: #FFFFFF; /* Màu chữ khi nút được nhấn */  
}
```

- Biểu đồ:

```
.chart-container {  
    display: flex;  
    flex-direction: row; /* Chuyển sang dạng cột */  
    gap: 20px; /* Khoảng cách giữa các cột */  
    width: 100%;  
    max-width: 1200px; /* Giới hạn chiều rộng */  
    margin: 0 auto; /* Canh giữa theo chiều ngang */  
    padding: 20px;  
}  
  
.chart-row {  
    flex: 1; /* Chia đều chiều rộng của các cột */  
}  
  
.chart {  
    height: 300px; /* Độ cao của biểu đồ */  
}
```

- Đối với màn hình:

- + Code dành cho màn hình có độ rộng nhỏ hơn 800px:

```
/* Responsive layout - when the screen is  
@media screen and (max-width: 800px) {  
    .leftcolumn, .rightcolumn {  
        width: 100%;  
        padding: 0;  
    }  
}
```

Thực tập Kiến trúc và Giao thức IoT

+ Code dành cho màn hình có độ rộng nhỏ hơn 400px:

```
/* Responsive layout - when the screen is
@media screen and (max-width: 400px) {
    .topnav a {
        float: none;
        width: 100%;
    }
}
```

2.3 JavaScript:

- Hàm ‘dongho()’ lấy thời gian hiện tại và hiển thị nó trong phần tử HTML có ID là “time”. Điều này tạo ra một hiển thị đồng hồ thời gian thực trong trang web của bạn, tự động cập nhật mỗi giây.

```
function dongho() {
    var time = new Date();
    var gio = time.getHours();
    var phut = time.getMinutes();
    var giay = time.getSeconds();
    if (gio < 10)
        gio = "0" + gio;
    if (phut < 10)
        phut = "0" + phut;
    if (giay < 10)
        giay = "0" + giay;
    document.getElementById("time").innerHTML = "Viet Nam (GMT+7) " + gio + ":" + phut + ":" + giay;
    setTimeout("dongho()", 1000);
};
```

- Hàm ‘ClickKitchen()’ chuyển hình nền và hiển thị phòng bếp khi được gọi. ‘ClickLivingRoom()’ và ClickBedRoom() cũng thực hiện tương tự, nhưng cho phòng khách và phòng ngủ. ClickVD() ẩn hình nền và hiển thị dữ liệu trực quan khi được gọi.

```
function ClickKitchen() {
    var roomElements = document.getElementsByClassName('room');
    document.getElementById('bgr_img').style.backgroundImage = "url('img/background-kitchen.jpg')";
    // Ẩn tất cả các phần tử có class 'room'
    for (var i = 0; i < roomElements.length; i++) {
        roomElements[i].style.display = 'none';
    }
    // Hiển thị phòng kitchen
    var kitchenElement = document.getElementById('kitchen');
    kitchenElement.style.display = 'block';
};
```

- Khởi tạo và kết nối Firebase với ứng dụng thông qua cấu hình ‘firebaseConfig’. Nó sử dụng các thông tin cần thiết như ‘apiKey’, ‘authDomain’, ‘databaseURL’, ‘projectId’, và các thông tin khác để xác định ứng dụng đang kết nối đến dịch vụ Firebase. ‘firebase.initializeApp()’ được sử dụng để khởi tạo kết nối với cấu hình đã cung cấp, trong khi ‘firebase.analytics()’ khởi tạo tính năng phân tích.

```
const firebaseConfig = {
    apiKey: "AIzaSyBWfuLCmEY65-V1aKI05ws9j8zp2S7xDI",
    authDomain: "smart-home-iot-5c4de.firebaseio.com",
    databaseURL: "https://smart-home-iot-5c4de.firebaseio.asia-southeast1.firebaseio.app",
    projectId: "smart-home-iot-5c4de",
    storageBucket: "smart-home-iot-5c4de.appspot.com",
    messagingSenderId: "129455497192",
    appId: "1:129455497192:web:12cf5e59c6d9699dd90e5d"
};
firebase.initializeApp(firebaseConfig);
firebase.analytics();
```

- Định nghĩa và liên kết các biến với các phần tử HTML khác nhau để hiển thị dữ liệu từ các phòng khác nhau trong Firebase.

Thực tập Kiến trúc và Giao thức IoT

+ Các biến như ‘nhietdoPK’, ‘dbReftempPK’, ‘doamPK’,... đều đại diện cho các phần tử trên giao diện HTML thông qua ‘getElementById’. Mỗi biến liên kết với một vị trí cụ thể trong cơ sở dữ liệu Firebase thông qua ‘firebase.database().ref().child(...)'’.

```
var nhietdoPK = document.getElementById('TemperatureLivingRoom');
var dbReftempPK = firebase.database().ref().child('Living room/Temperature');
var doamPK = document.getElementById('HumidityLivingRoom');
var dbRefhumipK = firebase.database().ref().child('Living room/Humidity');
var gasPK = document.getElementById('GasLivingRoom');
var dbRefgasPK = firebase.database().ref().child('Living room/Gas');
```

+ Các biến kết nối với trạng thái của các thiết bị trong các phòng khác nhau trong cơ sở dữ liệu Firebase. Mỗi biến ‘dbRefState...’ liên kết với một nút cụ thể trong cơ sở dữ liệu Firebase.

```
var dbRefStateAirConditionLivingRoom = firebase.database().ref().child('Living room/Air_conditional');
var dbRefStateLampLivingRoom = firebase.database().ref().child('Living room/Bulb');
var dbRefStateTelevisionLivingRoom = firebase.database().ref().child('Living room/TV');

var dbRefStateAirConditionBedroom = firebase.database().ref().child('Bedroom/Air_Conditional');
var dbRefStateLampBedroom = firebase.database().ref().child('Bedroom/Bulb');
var dbRefStateMusicBedroom = firebase.database().ref().child('Bedroom/Music');

var dbRefStatePanKitchen = firebase.database().ref().child('Kitchen/Fan');
var dbRefStateLampKitchen = firebase.database().ref().child('Kitchen/Bulb');
var dbRefStateCaxHoodKitchen = firebase.database().ref().child('Kitchen/Hood');
```

+ Các biến ‘State...’ được sử dụng để theo dõi trạng thái của các thiết bị trong các phòng khác nhau. Chúng có giá trị ban đầu là “0”.

```
var StateAirConditionLivingRoom = "0";
var StateLampLivingRoom = "0";
var StateTelevisionLivingRoom = "0";

var StateAirConditionBedroom = "0";
var StateLampBedroom = "0";
var StateMusicBedroom = "0";

var StatePanKitchen = "0";
var StateLampKitchen = "0";
var StateCaxHoodKitchen = "0";
```

+ Sau khi liên kết được thiết lập, các sự kiện ‘on('value', ...)’ được kích hoạt khi giá trị của dữ liệu trong cơ sở thay đổi, và giá trị mới sẽ được gán cho các phần tử tương ứng trên giao diện thông qua ‘innerText’.

```
dbReftempPK.on('value', snap => nhietdoPK.innerText = snap.val());
dbRefhumipK.on('value', snap => doamPK.innerText = snap.val());
dbRefgasPK.on('value', snap => gasPK.innerText = snap.val());

dbRefgasPN.on('value', snap => gasPN.innerText = snap.val());
dbReftempPN.on('value', snap => nhietdoPN.innerText = snap.val());
dbRefhumipPN.on('value', snap => doamPN.innerText = snap.val());

dbRefgasPB.on('value', snap => gasPB.innerText = snap.val());
dbReftempPB.on('value', snap => nhietdoPB.innerText = snap.val());
dbRefhumipPB.on('value', snap => doamPB.innerText = snap.val());
```

- Hàm ‘sync_Firebase()’ lấy dữ liệu thời gian thực từ Firebase và xử lý các trạng thái của các thiết bị trong phòng khách như điều hòa, đèn và TV. Ví dụ, dbRefStateTelevisionLivingRoom.on('value', function(snapshot) {...}); lắng nghe sự thay đổi trong trạng thái của điều hòa ở phòng khách. Khi có sự thay đổi, nó cập nhật trạng thái tương ứng của biến ‘StateTelevisionLivingRoom’ và thực hiện các hàm GVHD: Trương Quang Phúc

Thực tập Kiến trúc và Giao thức IoT

‘onTelevisionLivingRoom()’ hoặc ‘offTelevisionLivingRoom()’ tùy thuộc vào trạng thái nhận được từ Firebase. Các logic tương tự được áp dụng cho các thiết bị khác sử dụng 2 button điều khiển.

```
//Lấy dữ liệu Tivi Living Room
dbRefStateTelevisionLivingRoom.on('value', function(snapshot)
{
    StateTelevisionLivingRoom = snapshot.val();
    console.log(StateTelevisionLivingRoom);

    if (StateTelevisionLivingRoom == "1")
    {
        onTelevisionLivingRoom();
    }
    else
    {
        offTelevisionLivingRoom();
    }
}, function(error)
{
    | console.error('Error listening for changes in Firebase:', error);
});
```

- Lấy dữ liệu thời gian thực từ firebase nhưng đối với thiết bị điều khiển bằng slider: ‘dbRefStateAirConditionLivingRoom.on('value', function(snapshot) {...})’ lắng nghe sự thay đổi trong dữ liệu của điều hòa phòng khách trên Firebase. Khi có sự thay đổi, nó sẽ thực thi một hàm. Trong hàm được thực thi, ‘snapshot.val()’ trả về giá trị hiện tại của trạng thái điều hòa trong phòng khách. Nếu trạng thái là "1" (có thể là trạng thái bật), code sẽ thực hiện các hành động sau:

- + Gửi yêu cầu cập nhật nhiệt độ của phòng khách trong Firebase thành 16 độ.
- + Cập nhật giá trị của thanh trượt và hiển thị giá trị nhiệt độ là 16.
- + Thay đổi hình ảnh của điều hòa phòng khách để hiển thị trạng thái bật.

- Nếu trạng thái không phải là "1" (có thể là trạng thái tắt hoặc khác), code sẽ thực hiện các hành động sau:

- + Gửi yêu cầu cập nhật nhiệt độ của phòng khách trong Firebase thành "đè bé do lại nha".
- + Cập nhật giá trị của thanh trượt và hiển thị thông báo "Off".

Thực tập Kiến trúc và Giao thức IoT

```
//Lấy dữ liệu điều hòa Livingroom
dbRefStateAirConditionLivingRoom.on('value', function(snapshot)
{
    StateAirConditionLivingRoom = snapshot.val();
    console.log(StateAirConditionLivingRoom);
    if (StateAirConditionLivingRoom == "1")
    {
        firebase.database().ref().child('Living room/Temperature').set("16")
        airConditionerSlider.value = 16;
        sliderValue.innerText = "16";
        document.getElementById('dieuhoaphongkhach').src = "img/airon.png";
    }
    else
    {
        firebase.database().ref().child('Living room/Temperature').set("đè bê do lại nha")
        airConditionerSlider.value = 15;
        sliderValue.innerText = "Off";
    }
}, function(error)
{
    console.error('Error listening for changes in Firebase:', error);
});
```

- Lấy dữ liệu thời gian thực từ firebase đối với thiết bị được điều khiển bằng 1 button duy nhất: ‘dbRefStateLampLivingRoom.on('value', function(snapshot) {...})’ lắng nghe sự thay đổi trong dữ liệu của đèn trong phòng khách trên Firebase. Khi có sự thay đổi, nó thực thi một hàm. Nếu trạng thái là "1" (có thể là trạng thái bật), thực hiện việc thay đổi hình ảnh của đèn trong phòng khách để hiển thị trạng thái bật và cập nhật text của button điều khiển thành “On”. Nếu trạng thái không phải là "1" (trạng thái khác), thực hiện việc thay đổi hình ảnh của đèn trong phòng khách để hiển thị trạng thái tắt và cập nhật text của button điều khiển thành “Off”.

```
//Lấy dữ liệu đèn Living Room
dbRefStateLampLivingRoom.on('value', function(snapshot)
{
    StateLampLivingRoom = snapshot.val();
    console.log(StateLampLivingRoom);

    if (StateLampLivingRoom == "1")
    {
        document.getElementById('denphongkhach').src = "img/lamp2.png";
        lampButton.innerText = "On";
    }
    else
    {
        document.getElementById('denphongkhach').src = "img/lamp1.png";
        lampButton.innerText = "Off";
    }
}, function(error)
{
    console.error('Error listening for changes in Firebase:', error);
});
```

- Code điều khiển slider của điều hòa phòng khách và cập nhật dữ liệu liên quan đến nhiệt độ vào Firebase khi slider thay đổi giá trị.

+ ‘airConditionerSlider.addEventListener('input', function () {...})’ lắng nghe sự thay đổi trong giá trị của slider khi người dùng thao tác. Khi slider thay đổi, nó thực thi một hàm.

Thực tập Kiến trúc và Giao thức IoT

+ Trong hàm được thực thi, ‘parseInt(this.value)’ chuyển đổi giá trị mới của slider thành một số nguyên để thực hiện so sánh và xử lý. Nếu giá trị mới là 15 (được hiểu là trạng thái "Off" của điều hòa) thì thực hiện:

- Thay đổi hình ảnh của điều hòa để hiển thị trạng thái "Off".
- Cập nhật trạng thái điều hòa trong phòng khách trên Firebase thành "0" (tắt).
- Cập nhật giá trị hiển thị của slider thành "Off".
- Cập nhật thông tin về nhiệt độ trong phòng khách trên Firebase thành "đè bé đeo lại nha".

```
//SLIDER AIR CONDITIONER LIVING ROOM
var airConditionerSlider = document.getElementById('airConditionerSlider');
var sliderValue = document.getElementById('sliderValue');
var previousSliderValue = 0; // Lưu trữ giá trị trước đó của slider
// Xử lý khi slider thay đổi giá trị
airConditionerSlider.addEventListener('input', function ()
{
    var newTemperatureValue = parseInt(this.value); // Chuyển đổi giá trị sang kiểu số nguyên
    if (newTemperatureValue === 15)
    {
        document.getElementById('dieuhoaphongkhach').src = "img/airoff.png";
        dbRefStateAirConditionLivingRoom.set("0");
        sliderValue.innerText = "Off"; // Nếu giá trị là 0, hiển thị là "Off"
        // Cập nhật giá trị "đè bé đeo lại nha" cho Living room Temperature khi slider ở trạng thái "off"
        firebase.database().ref().child('Living room/Temperature').set("đè bé đeo lại nha")
        .then(function ()
        {
            console.log("Cập nhật Living room Temperature thành công!");
        })
        .catch(function (error)
        {
            console.error("Cập nhật Living room Temperature thất bại:", error);
        });
    }
});
```

+ Nếu giá trị không phải là 15 (tức là trạng thái không phải "Off"):

- Thay đổi hình ảnh của điều hòa để hiển thị trạng thái bật.
- Cập nhật trạng thái điều hòa trong phòng khách trên Firebase thành "1" (bật).
- Hiển thị giá trị nhiệt độ mới trên slider.
- Cập nhật thông tin về nhiệt độ trong phòng khách trên Firebase theo giá trị mới của slider.

```
else
{
    document.getElementById('dieuhoaphongkhach').src = "img/airon.png";
    dbRefStateAirConditionLivingRoom.set("1");
    sliderValue.innerText = newTemperatureValue; // Hiển thị giá trị trên slider
    // Cập nhật giá trị nhiệt độ lên Firebase từ 16 đến 32 khi slider không ở trạng thái "off"
    firebase.database().ref().child('Living room/Temperature').set(newTemperatureValue)
    .then(function ()
    {
        console.log("Cập nhật nhiệt độ thành công!");
    })
    .catch(function (error)
    {
        console.error("Cập nhật nhiệt độ thất bại:", error);
    });
}
// Lưu trữ giá trị hiện tại của slider cho lần lặp tiếp theo
previousSliderValue = newTemperatureValue;
});
```

- Hàm LampLivingroom() khi được gọi sẽ lấy giá trị hiện tại của đèn từ cơ sở dữ liệu Firebase, đảo ngược trạng thái hiện tại của đèn và cập nhật trạng thái mới vào cơ sở dữ

Thực tập Kiến trúc và Giao thức IoT

liệu Firebase. Ngoài ra, nó cũng cập nhật nội dung của nút điều khiển đèn trên giao diện người dùng để phản ánh trạng thái mới của đèn ("On" hoặc "Off").

```
//ONE BUTTON LAMP LIVING ROOM
function LampLivingroom()
{
    // Lấy giá trị hiện tại của đèn từ Firebase
    dbRefStateLampLivingRoom.once('value', function(snapshot)
    {
        var lampState = snapshot.val(); // Lấy giá trị trạng thái hiện tại của đèn
        // Đảo ngược trạng thái của đèn
        var newLampState = lampState === "0" ? "1" : "0";
        // Cập nhật giá trị mới vào Firebase
        dbRefStateLampLivingRoom.set(newLampState)
        .then(function()
        {
            console.log("Trạng thái đèn đã được cập nhật thành công!");
            lampButton.innerText = newLampState === "1" ? "On" : "Off";
        })
        .catch(function(error)
        {
            console.error("Lỗi khi cập nhật trạng thái đèn:", error);
        });
    });
}
```

- Hàm onTelevisionLivingRoom() được kích hoạt khi trạng thái của điều hòa phòng khách chuyển sang "bật". Đối với các hàm tắt/bật khác cũng tương tự. Khi được kích hoạt, hàm này thực hiện các hành động sau:

+ Đầu tiên, nó cập nhật trạng thái điều hòa trong cơ sở dữ liệu Firebase thông qua dbRefStateTelevisionLivingRoom.set(StateTelevisionLivingRoom). Sau đó, nó kiểm tra xem việc cập nhật thành công hay không và ghi log tương ứng.

+ Hàm thay đổi hình ảnh của đối tượng 'tiviphongkhach' thành hình ảnh "tivi bật" (img/TVon.png) và thêm class "active" cho một phần tử có ID 'ontv1'.

```
function onTelevisionLivingRoom()
{
    StateTelevisionLivingRoom = "1";
    dbRefStateTelevisionLivingRoom.set(stateTelevisionLivingRoom)
    .then(function() {
        console.log("Cập nhật giá trị thành công!");
    })
    .catch(function(error) {
        console.error("Cập nhật giá trị thất bại:", error);
    });

    document.getElementById('tiviphongkhach').src = "img/TVon.png";
    document.getElementById('ontv1').classList.add("active");
}
```

- Code xử lý khi click vào các nút top navigation bar: Khi người dùng click vào một nút tab, tất cả các nút khác sẽ mất lớp "active" và chỉ có nút được click sẽ có lớp "active", từ đó giúp xác định rõ nút nào đang được chọn.

```
// Lấy tất cả các nút tab
const topnavButtons = document.querySelectorAll(".topnav button");

// Lặp qua từng nút tab
topnavButtons.forEach((button) => {
    button.addEventListener("click", function () {
        // Loại bỏ lớp "active" từ tất cả các nút tab
        topnavButtons.forEach((btn) => {
            btn.classList.remove("active");
        });

        // Thêm lớp "active" cho nút tab đang được chọn
        button.classList.add("active");
    });
});
```

2.4. Chart:

- Khởi tạo dữ liệu và cấu trúc cho biểu đồ sẽ được vẽ để theo dõi các thông số nhiệt độ, độ ẩm và nồng độ khí gas của phòng Living Room.

+ ‘temperatureDataPK’, ‘humidityDataPK’, và ‘gasDataPK’ là các mảng dữ liệu được sử dụng để lưu trữ các giá trị tương ứng của nhiệt độ, độ ẩm và khí gas.

```
// Hiển thị biểu đồ Livingroom
var temperatureDataPK = [];
var humidityDataPK = [];
var gasDataPK = [];
var timeLabelsPK = [];
```

+ ‘timeLabelsPK’: mảng thời gian sẽ được sử dụng như là nhãn trên trục x của biểu đồ.

+ ‘ctxPK’: đối tượng mà Canvas sẽ vẽ biểu đồ lên, lấy thông tin dựa vào id của element 'LivingroomChart'.

+ ‘dataPK’: chứa cấu trúc dữ liệu cho biểu đồ, bao gồm:

+ ‘labels’: Lưu trữ mảng thời gian (timeLabelsPK) để làm nhãn cho trục x.

+ ‘datasets’: Mảng chứa các dữ liệu của từng đường line trên biểu đồ. Mỗi dataset có các thông số như: ‘label’ - tên của dữ liệu, ở đây là 'Temperature', 'Humidity', 'Gas'; ‘data’ - mảng chứa các giá trị tương ứng của nhiệt độ, độ ẩm và khí gas; ‘borderColor’ - màu sắc đường viền của line; ‘borderWidth’ - Độ dày của đường line; ‘fill’ - xác định liệu line có được tô màu hay không.

Thực tập Kiến trúc và Giao thức IoT

```
// Sử dụng mảng thời gian làm nhãn cho biểu đồ
var ctxPK = document.getElementById('LivingroomChart').getContext('2d');
var dataPK = {
  labels: timeLabelsPK,
  datasets: [
    {
      label: 'Temperature',
      data: temperatureDataPK,
      borderColor: 'red',
      borderWidth: 2,
      fill: false,
    },
    {
      label: 'Humidity',
      data: humidityDataPK,
      borderColor: 'blue',
      borderWidth: 2,
      fill: false,
    },
    {
      label: 'Gas',
      data: gasDataPK,
      borderColor: 'green',
      borderWidth: 2,
      fill: false,
    },
  ],
};
```

- Hàm cập nhật biểu đồ:

```
// Hàm cập nhật biểu đồ
3 references
function updateChartPK() {
  LivingroomChart.update();
}
```

- ‘dbRefTempPK.on('value', ...)' : Lắng nghe sự thay đổi của giá trị nhiệt độ trong phòng Living Room từ Firebase. Mỗi khi có dữ liệu mới, hàm ‘callback’ được gọi. Dữ liệu mới sẽ được đưa vào mảng ‘temperatureDataPK’, cùng với thời gian tương ứng vào mảng ‘timeLabelsPK’. Nếu mảng đạt độ dài lớn hơn 10 (chỉ lưu trữ 10 điểm dữ liệu gần nhất), các phần tử cũ sẽ được loại bỏ bằng cách sử dụng ‘shift()’, sau đó gọi hàm ‘updateChartPK()’ để cập nhật biểu đồ. Thực hiện tương tự cho độ ẩm và gas.

Thực tập Kiến trúc và Giao thức IoT

```
// Lắng nghe dữ liệu từ Firebase và cập nhật biểu đồ
dbRefTempPK.on('value', (snapshot) => {
  const time = new Date().toLocaleTimeString();
  timeLabelsPK.push(time);
  temperatureDataPK.push(snapshot.val());
  if (temperatureDataPK.length > 10) {
    temperatureDataPK.shift();
    timeLabelsPK.shift();
  }
  updateChartPK();
});

dbRefhumiPK.on('value', (snapshot) => {
  humidityDataPK.push(snapshot.val());
  if (humidityDataPK.length > 10) {
    humidityDataPK.shift();
  }
  updateChartPK();
});

dbRefgasPK.on('value', (snapshot) => {
  gasDataPK.push(snapshot.val());
  if (gasDataPK.length > 10) {
    gasDataPK.shift();
  }
  updateChartPK();
});
```

- Vẽ biểu đồ dạng đường cho Living Room, với cấu hình hiển thị thông tin nhiệt độ, độ ẩm, nồng độ khí gas theo thời gian:

+ optionsPK: Chứa cấu hình cho biểu đồ, bao gồm các thuộc tính:

Thuộc tính	Mô tả
responsive	Cho phép biểu đồ tự động thay đổi kích thước để phù hợp với kích thước của phần tử cha.
maintainAspectRatio	Duy trì tỉ lệ khung hình ban đầu khi thay đổi kích thước.
scales	Cấu hình các trục của biểu đồ.

+ LivingroomChart: Tạo một biểu đồ mới từ thư viện Chart.js với các thông số sau:

Thuộc tính	Mô tả
ctxPK	Đối tượng canvas để vẽ biểu đồ.
type	Loại biểu đồ, trong trường hợp này là biểu đồ đường ('line').
data	Dữ liệu của biểu đồ, được định nghĩa từ dataPK (bao gồm nhãn và dữ liệu từ Firebase).
options	Cấu hình của biểu đồ, được định nghĩa từ optionsPK.

```
// Cấu hình biểu đồ
const optionsPK = {
  responsive: true,
  maintainAspectRatio: true,
  scales: {
    x: {
      display: true,
      title: {
        display: true,
        text: 'Livingroom',
      },
    },
    y: {
      display: true,
      title: {
        display: true,
        text: 'Giá trị',
      },
    },
  },
};

const LivingroomChart = new Chart(ctxPK, {
  type: 'line',
  data: dataPK,
  options: optionsPK,
});
```

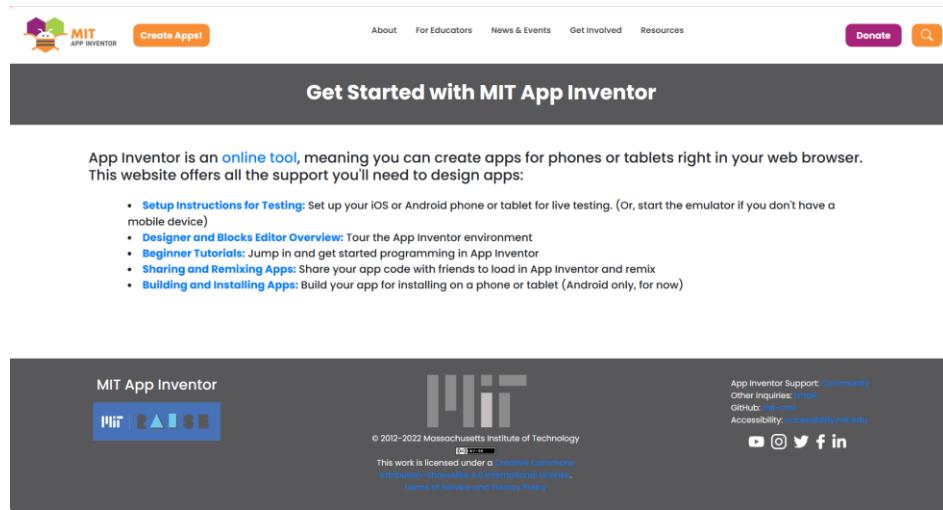
- Làm tương tự với Bedroom và Kitchen.

3. Giao diện điều khiển MIT app

- Nhóm chọn “MIT App Inventor” để thiết kế giao diện ứng dụng điều khiển “Smart Home” trên các thiết bị di động vì tính dễ sử dụng và tương thích được với các thiết bị chạy hệ điều hành Android.

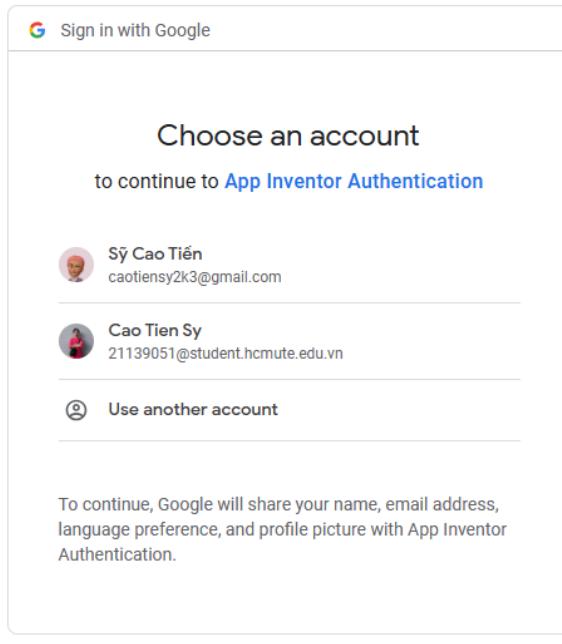
- Để bắt đầu thiết kế giao diện ứng dụng điều khiển chúng ta truy cập vào đường dẫn: <https://appinventor.mit.edu/>

Thực tập Kiến trúc và Giao thức IoT



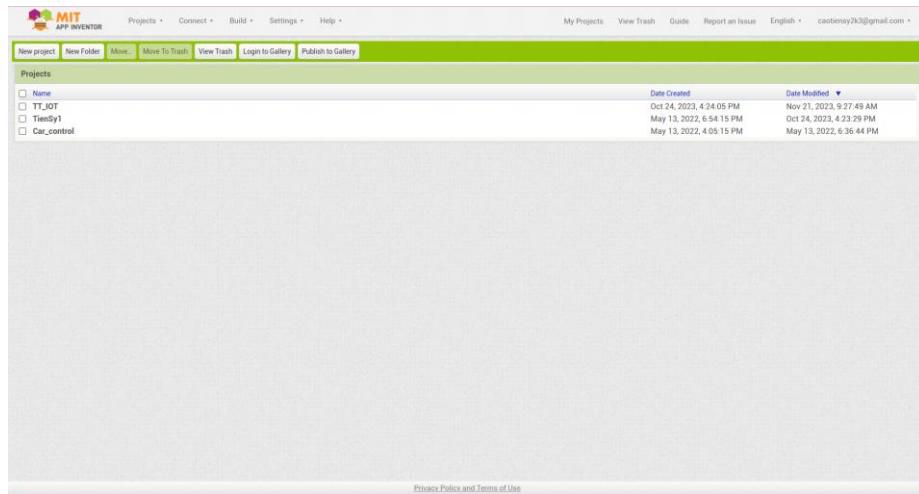
Hình 8: Trang chủ của MIT App Inventor

- Ta sẽ bấm vào nút Create Apps! ở góc trái phía trên màn hình. Sau đó đăng nhập tài khoản Google.



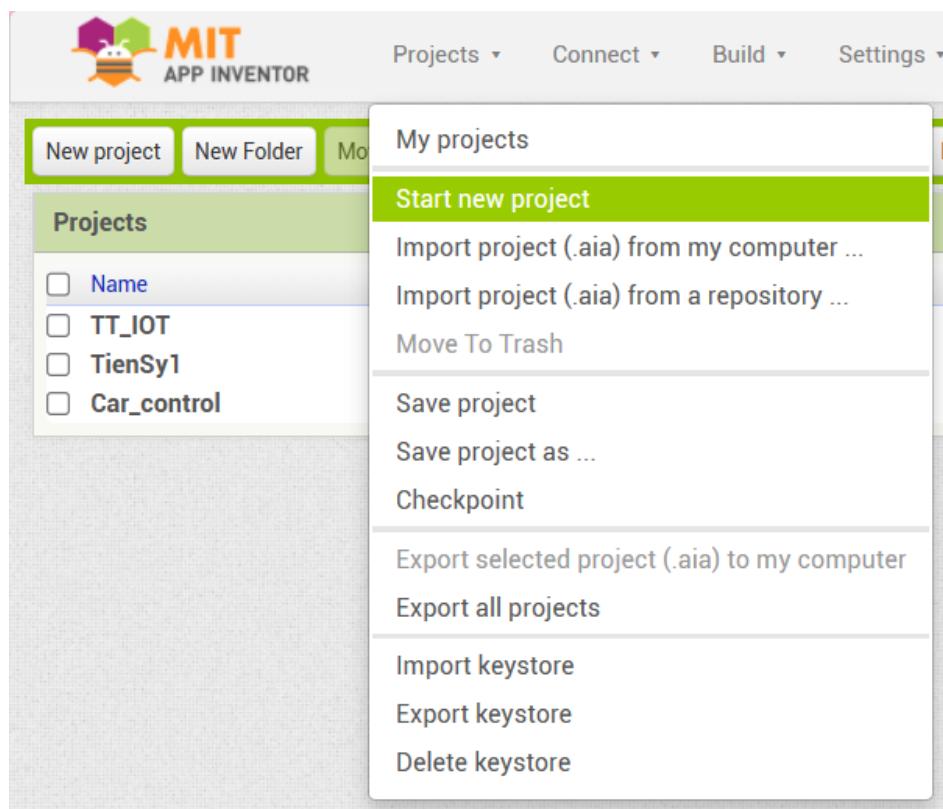
Hình 9: Đăng nhập bằng tài khoản Google

Thực tập Kiến trúc và Giao thức IoT



Hình 10: Giao diện Project MIT App Inventor

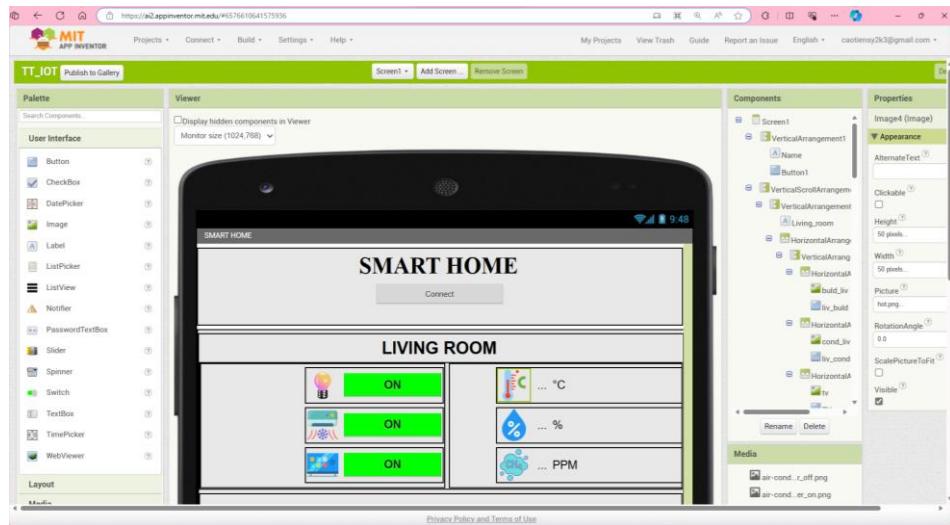
- Để tạo dự án mới, ta nhấn chọn Project => Start new project.



Hình 11: Tạo dự án mới

- Đối với các thiết bị di động hiện nay chúng ta sẽ chọn kích thước để kiểm kê là Monitor size để thoải mái không gian hơn trong quá trình thiết kế.

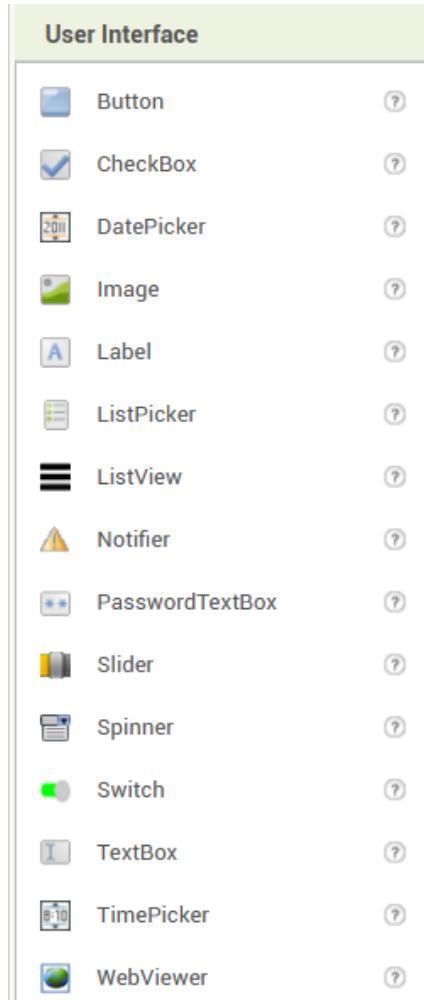
Thực tập Kiến trúc và Giao thức IoT



Hình 12: Giao diện thiết kế giao diện ứng dụng

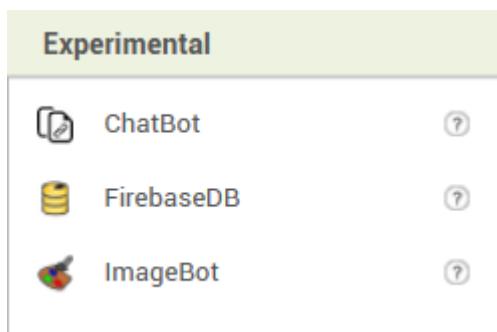
- Để đặt những giao tiếp với người dùng chúng ta sẽ phân vùng cho chúng bằng cách bấm vào Layout và chọn kiểu phân vùng (tùy theo thiết kế) rồi kéo thả vào màn hình thiết kế chính.

- Thiết kế của nhóm sẽ có kiểu sắp xếp theo chiều dọc (VerticalArrangement) vì các nội dung hiển thị sẽ được sắp xếp theo thứ tự từ trên xuống.



Thực tập Kiến trúc và Giao thức IoT

- Mỗi phòng sẽ là 1 vùng được sắp xếp theo chiều dọc để hiển thị tên phòng và hiển thị trạng thái thiết bị và thông số môi trường.
- Chúng ta kéo thả đối tượng “Label” trong mục “User Interface” vào trong vùng được chia cho mỗi phòng và thuộc tính “Text” để đặt tên cho mỗi phòng.
- Các trạng thái thiết bị và thông số môi trường sẽ được hiển thị trong vùng được sắp xếp theo chiều ngang (HorizontalArrangement) trong đó có 2 vùng được sắp xếp theo chiều dọc, 1 vùng sẽ hiển thị trạng thái các thiết bị và vùng còn lại sẽ hiển thị thông số môi trường của phòng.
 - Trong vùng được chia để hiển thị trạng thái thiết bị, mỗi thiết bị sẽ được hiển thị trong vùng sắp xếp theo chiều ngang bao gồm: “Image”: ảnh hiển thị thiết bị và “Button”: điều khiển và hiển thị trạng thái thiết bị. Tương tự với các thiết bị còn lại.
 - Đối với vùng được chia để hiển thị thông số môi trường, ta cũng sẽ làm giống như trạng thái của thiết bị nhưng bao gồm: “Image”: ảnh đại diện cho môi trường và “Label”: hiển thị thông số. Tương tự với các thông số môi trường khác.
 - Sau khi đã xây dựng giao diện cho ứng dụng xong, Nhóm kéo thả “FirebaseDB” trong mục “Experimental” để liên kết với Firebase.

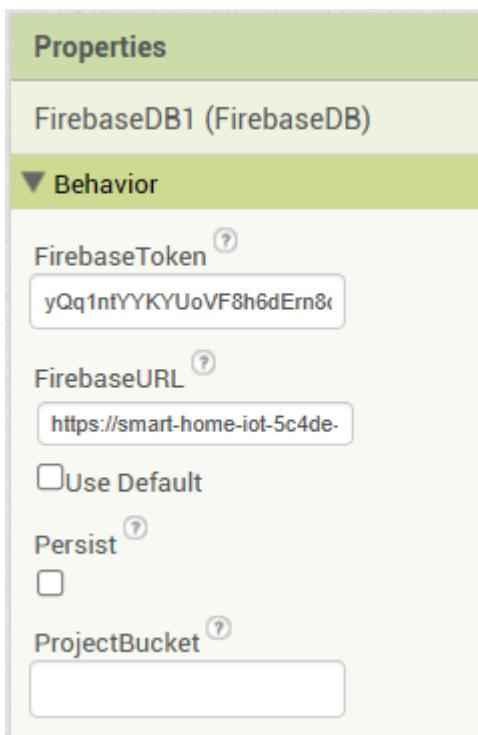


- Chúng ta chọn vào “FirebaseDB1” để bắt đầu chỉnh sửa cấu hình.

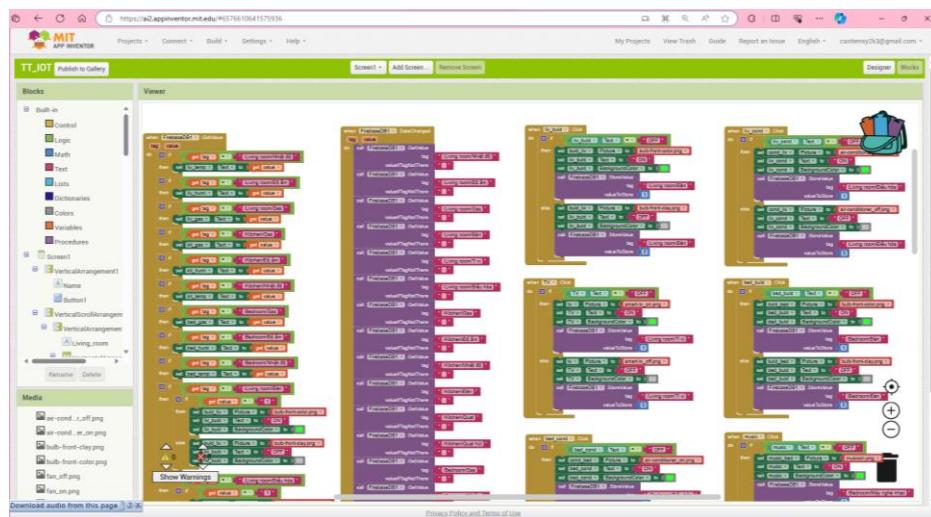


- Chúng ta thêm FirebaseAuthToken và FirebaseAuthURL của mình vào 2 mục tương ứng.

Thực tập Kiến trúc và Giao thức IoT

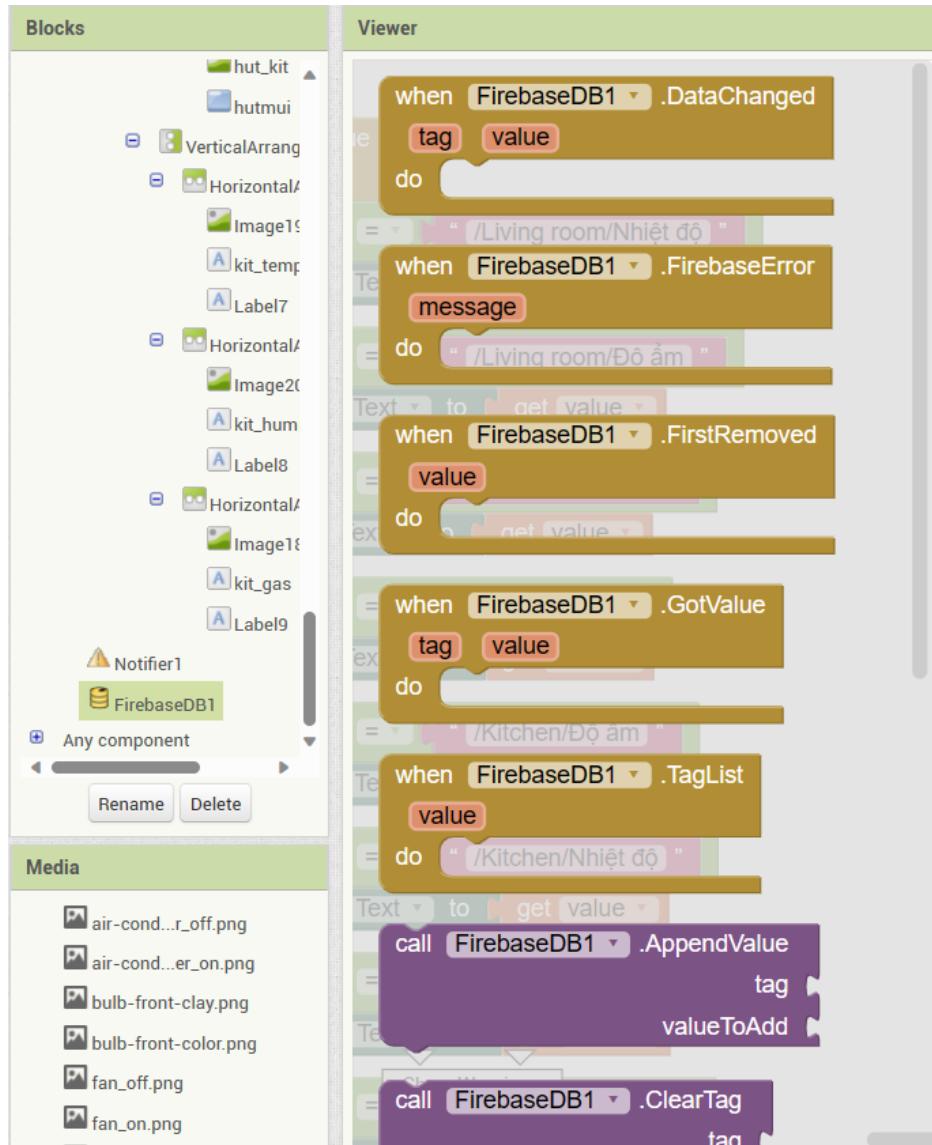


- Sau đó chúng ta sẽ xây dựng chức năng cho từng đối tượng mà chúng ta đã xây dựng ở trong trang “Blocks” nằm ở phía trên góc phải màn hình.

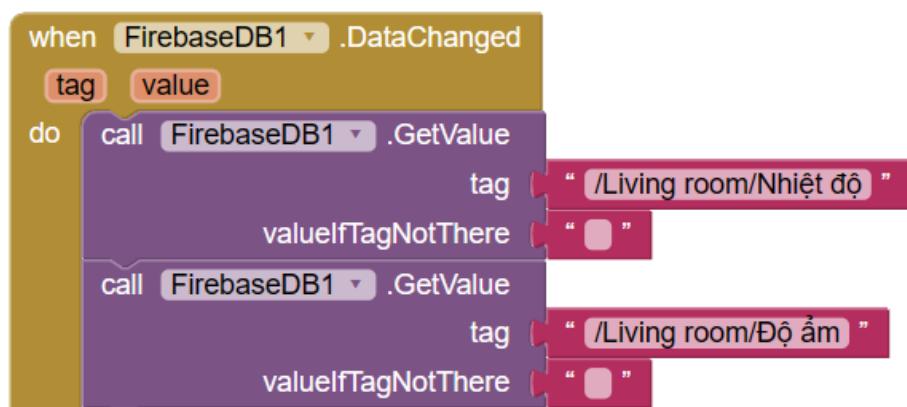


Thực tập Kiến trúc và Giao thức IoT

- Trong mục “Blocks” chúng ta tìm và chọn “FirebaseDB1” sẽ hiển thị ra các “block” như sau:

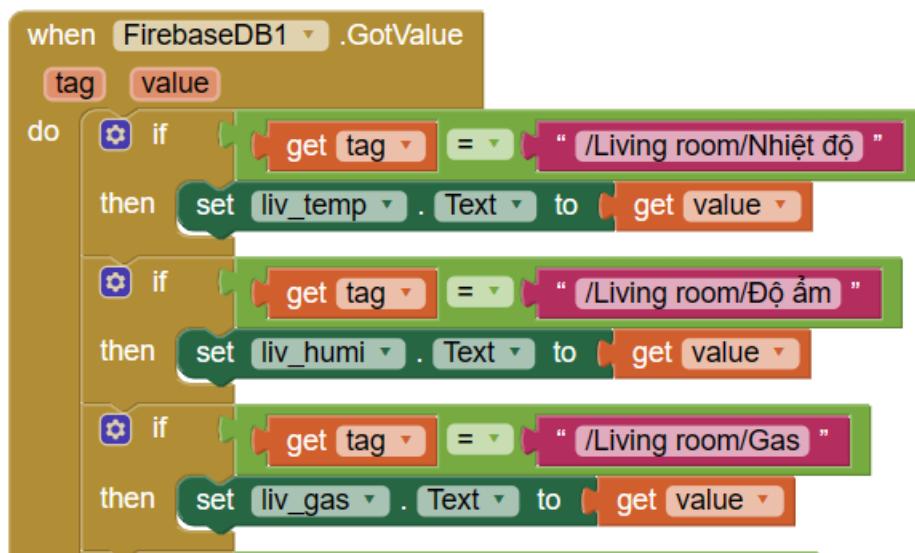


- Nhóm sẽ chọn .DataChanged để cập nhật các thông tin nếu Firebase có thay đổi.
- Thêm các thẻ .GetValue đặt giá trị “tag” là đường dẫn của thông tin được cài đặt trên Firebase để lấy giá trị của từng đối tượng.

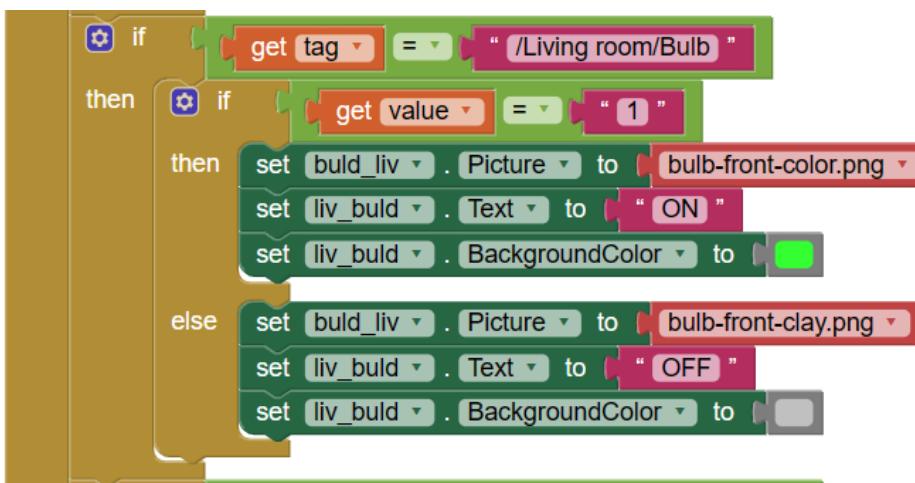


Thực tập Kiến trúc và Giao thức IoT

- Chọn .GetValue để hiển thị những thông tin đã thay đổi với người dùng.
- Đối với các thông số môi trường chúng ta sẽ sử dụng câu lệnh if để lấy thông tin của thông số tại phòng thay đổi và hiển thị thông số với người dùng như sau:



- Đối với trạng thái thiết bị, chúng ta sẽ cần làm nhiều hơn do tính chất của nó.
- Ta sử dụng câu lệnh if để xác định được đối tượng có thay đổi hay không. Nếu đối tượng thay đổi:
 - + Trường hợp 1: đang bật (được đặt với giá trị là 1) thì chúng ta sẽ thay đổi hình ảnh đối tượng đang hoạt động, hiển thị trạng thái “Button” là “ON” và đổi màu thành màu xanh.
 - + Trường hợp 2: đang tắt (được đặt với giá trị là 0) thì chúng ta sẽ thay đổi hình ảnh đối tượng không hoạt động, hiển thị trạng thái “Button” là “OFF” và đổi màu thành xám.



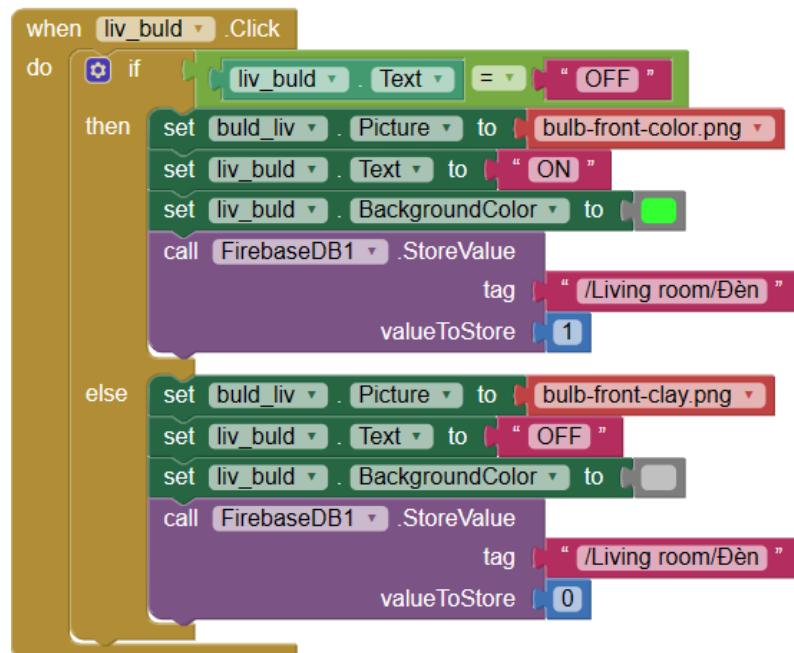
- Để người dùng tương tác với những đối tượng và sau đó gửi cá thay đổi lên Firebase ta làm như sau:

Thực tập Kiến trúc và Giao thức IoT

- Ví dụ với đối tượng bóng đèn tại phòng khách: Chúng ta sẽ tìm và chọn vào “Button” của bóng đèn tại phòng khách và kéo ra thẻ “when .Click”. Chúng ta sử dụng cấu trúc rẽ nhánh để thực hiện các thay đổi khi nhấn vào “Button”:

+ Nếu trạng thái trước khi nhấn của “Button” đang là “OFF” thì sau khi nhấn ta thay đổi hình ảnh của đèn sang trạng thái hoạt động và đổi trạng thái “Button” sang “ON” và màu thành màu xanh. Sau đó sẽ thay đổi trạng thái của đèn phòng khách trên Firebase thành đang bật “1”.

+ Ngược lại, trạng thái trước khi nhấn của “Button” là “ON” thì sau khi nhấn ta thay đổi hình ảnh của đèn sang trạng thái không hoạt động và đổi trạng thái “Button” sang “OFF” và màu thành xám. Sau đó sẽ thay đổi trạng thái của đèn phòng khách trên Firebase thành đang tắt “0”.

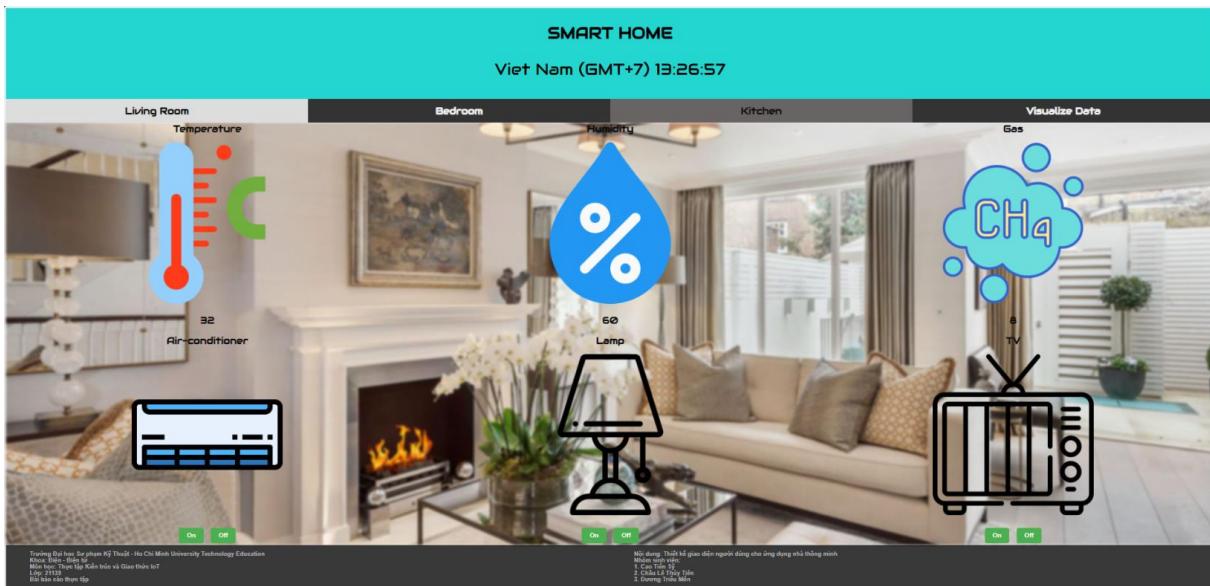


Thực tập Kiến trúc và Giao thức IoT

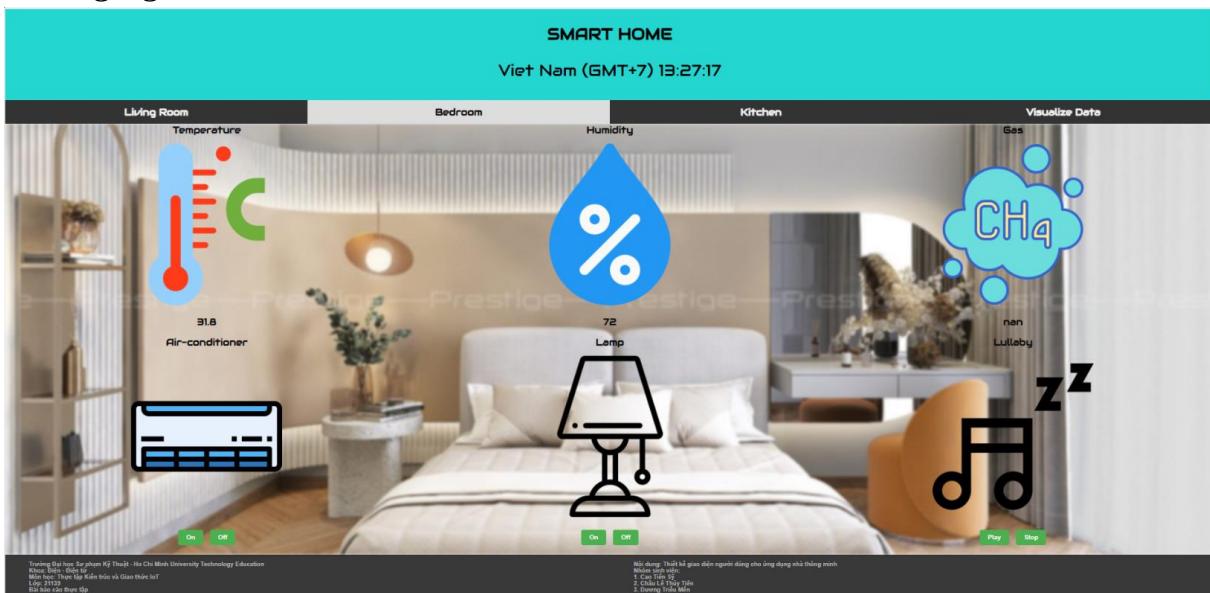
III. Trình bày kết quả

1. Kết quả web HTML

1.1. Phòng khách

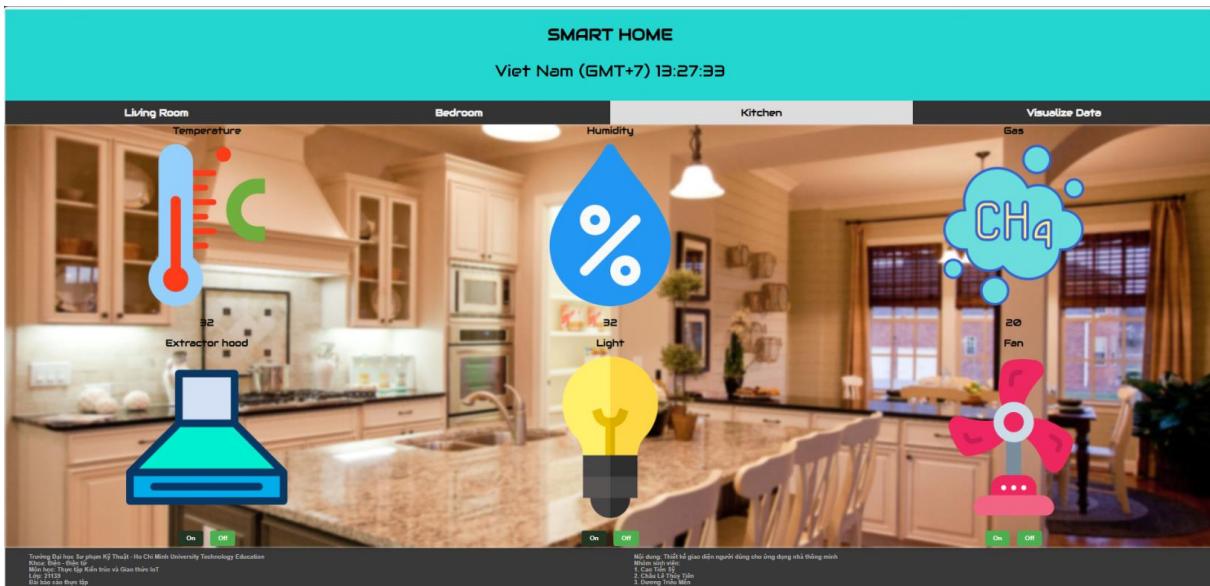


1.2. Phòng ngủ

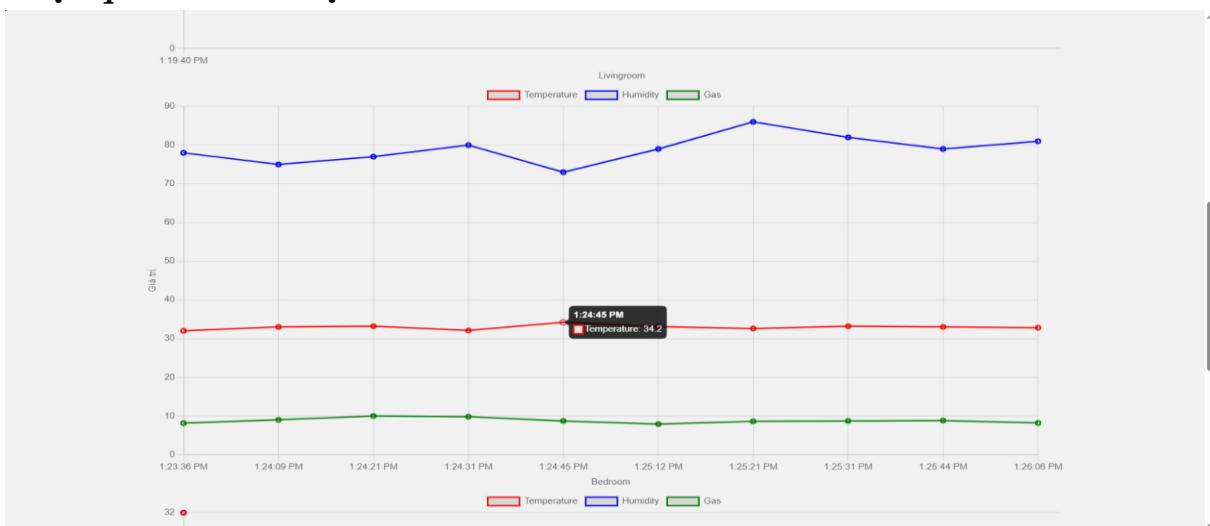


Thực tập Kiến trúc và Giao thức IoT

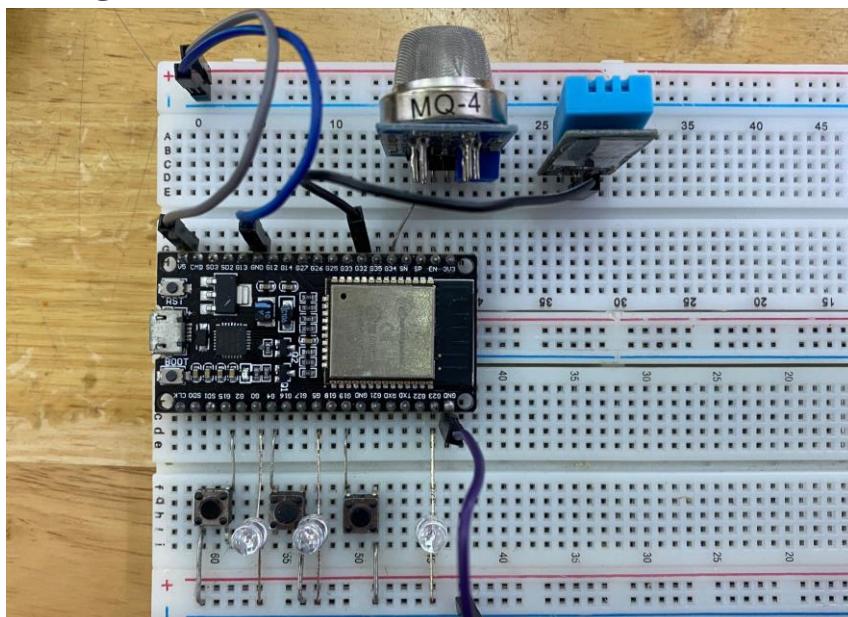
1.3. Phòng bếp



1.4. Trực quan hóa dữ liệu

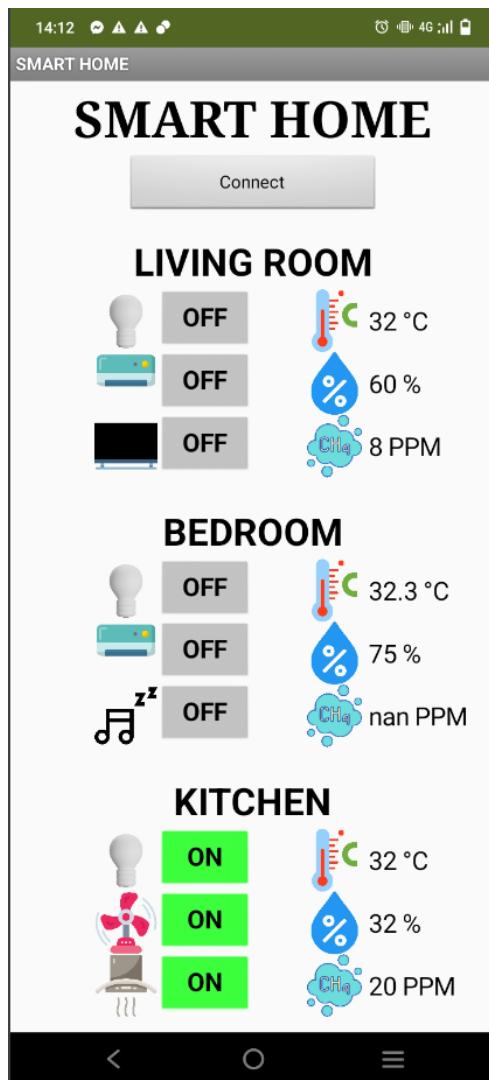


2. Kết quả phần cứng



GVHD: Trương Quang Phúc

3. Kết quả App MIT inventor



4. Video demo

- Video: Chạy các phần của đề tài.

IV. Tài liệu tham khảo

1. ESP32-DevKitC V4 Getting Started Guide. Truy cập vào ngày 10/10/2023 tại:
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>
2. Arduino, How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino. Truy cập ngày 10/10/2023 tại: <https://lastminuteengineers.com/mq2-gas-sensor-arduino-tutorial/>
3. miquel5612, MQSensorLib. Truy cập ngày 10/10/2023 tại:
<https://github.com/miguel5612/MQSensorLib>
4. ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE. Truy cập ngày 14/10/2023 tại: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>

B. Tìm hiểu HTML, CSS, JS và Google Firebase

I. HTML

1. HTML Headings

- HTML headings là các tiêu đề hoặc phụ đề mà bạn muốn hiển thị trên một trang web. <size 1> nên dùng cho tiêu đề chính, sau đó là <size 2> ít quan trọng hơn, tiếp đến là <size 3> và cứ tiếp tục như vậy.

Code	Result
<!DOCTYPE html>	Size 1
<html>	Size 2
<body>	Size 3
<h1>Size 1</h1>	Size 4
<h2>Size 2</h2>	Size 5
<h3>Size 3</h3>	Size 6
<h4>Size 4</h4>	
<h5>Size 5</h5>	
<h6>Size 6</h6>	
</body>	
</html>	

- Mỗi tiêu đề HTML có kích thước mặc định. Tuy nhiên, bạn có thể chỉ định kích thước cho bất kỳ tiêu đề nào bằng thuộc tính style, sử dụng thuộc tính font-size trong CSS:

Code	Result
<!DOCTYPE html>	Size 1
<html>	
<body>	
<h1 style="font-size:70px;">Size 1</h1>	
<p>Bạn có thể thay đổi kích thước tiêu đề bằng cách sử dụng font-size.</p>	Bạn có thể thay đổi kích thước tiêu đề bằng cách sử dụng font-size.
</body>	
</html>	

2. HTML Paragraphs

2.1. HTML Paragraphs

- Một đoạn văn luôn bắt đầu trên một dòng mới và thường là một khối văn bản.
- Thủ HTML <p> xác định một đoạn văn.

Thực tập Kiến trúc và Giao thức IoT

- Một đoạn văn luôn bắt đầu trên một dòng mới và các trình duyệt tự động thêm một số khoảng trắng (margin) trước và sau đoạn văn.

Code	Result
<!DOCTYPE html> <html> <body> <p>Paragraph 1.</p> <p>Paragraph 2.</p> <p>Paragraph 3.</p> </body> </html>	Paragraph 1. Paragraph 2. Paragraph 3.

2.2. HTML display:

- Màn hình lớn hoặc nhỏ, và cửa sổ được thay đổi kích thước sẽ tạo ra kết quả khác nhau. Với HTML, bạn không thể thay đổi cách hiển thị bằng cách thêm khoảng trắng hoặc dòng trống thừa vào mã HTML của bạn.

Code	Result
<!DOCTYPE html> <html> <body> <p> Đoạn văn này chứa rất nhiều dòng và khoảng trắng trong code, nhưng trình duyệt bỏ qua nó. </p> </body> </html>	Đoạn văn này chứa rất nhiều dòng và khoảng trắng trong code, nhưng trình duyệt bỏ qua nó.

2.3. HTML Horizontal Rules:

- Thủ <hr> xác định một phần ngắn chủ đề trong trang HTML, và thường được hiển thị như một đường kẻ ngang.
- Thủ <hr> được sử dụng để tách nội dung (hoặc định nghĩa một sự thay đổi) trong trang HTML.
- Thủ <hr> là một thủ trống, điều này có nghĩa là nó không có thủ kết thúc.

Code	Result

```
<!DOCTYPE html>
<html>
<body>
<h1>This is heading 1</h1>
<hr>
<h2>This is heading 2</h2>
<hr>
</body>
</html>
```

This is heading 1

This is heading 2

2.4. HTML Line Breaks:

- Sử dụng `
` nếu bạn muốn có một ngắt dòng (dòng mới) mà không bắt đầu một đoạn văn mới.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>This is
a paragraph
with line breaks.</p> </body> </html></pre>	<p>This is a paragraph with line breaks.</p>

2.5. The Poem Problem:

- Thơ sẽ chỉ hiển thị trên 1 dòng.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p> Nam quốc sơn hà Nam đế cư Tiết nhiên định phận tại thiên thư Như hà nghịch lỗ lai xâm phạm Nhữ đắng hành khan thủ bại hư. </p> </body> </html></pre>	<p>Nam quốc sơn hà Nam đế cư Tiết nhiên định phận tại thiên thư Như hà nghịch lỗ lai xâm phạm Nhữ đắng hành khan thủ bại hư.</p>

2.6. Giải pháp của ‘Poem problem’: `<pre>`

- Thủ `<pre>` trong HTML xác định văn bản được định dạng trước.
- Văn bản bên trong thẻ `<pre>` sẽ được hiển thị bằng một kiểu chữ có độ rộng cố định (thường là Courier), và nó bảo tồn cả khoảng trắng và ngắt dòng.

Code	Result

<pre><!DOCTYPE html> <html> <body> <pre> Nam quốc sơn hà Nam đế cư Tiệt nhiên định phận tại thiên thư Như hà nghịch lỗ lai xâm phạm Nhữ đẳng hành khan thủ bại hư. </pre> </body> </html></pre>	<p>Nam quốc sơn hà Nam đế cư Tiệt nhiên định phận tại thiên thư Như hà nghịch lỗ lai xâm phạm Nhữ đẳng hành khan thủ bại hư.</p>
---	---

3. HTML Styles

- Thuộc tính style HTML được sử dụng để thêm các kiểu cho một phần tử, chẳng hạn như màu sắc, phông chữ, kích thước và nhiều thuộc tính khác.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>normal</p> <p style="color: red;">red</p> <p style="color: blue;">blue</p> <p style="font-size:40px;">big</p> </body> </html></pre>	<p>normal</p> <p>red</p> <p>blue</p> <p>big</p>

- Thiết lập kiểu cho một phần tử HTML có thể được thực hiện bằng cách sử dụng thuộc tính "style". Cú pháp như sau: `tagname style="property:value;">`

3.1. Background Color:

- Thuộc tính CSS "background-color" xác định màu nền cho một phần tử HTML.

Code	Result
<pre><!DOCTYPE html> <html> <body style="background-color:lightgreen;"> <h1>This is a heading</h1> <p>This is a paragraph.</p> </body> </html></pre>	<p>This is a heading</p> <p>This is a paragraph.</p>

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:pink;">This is a heading</h1>
<p style="background-color:lightblue;">This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

3.2. Các thuộc tính khác

- "color" để đặt màu chữ.
- "font-family" để đặt font chữ.
- "font-size" để đặt kích thước chữ.
- "text-align" để căn chỉnh văn bản.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1 style="color:blue;">This is a heading</h1> <p style="color:red;">This is a paragraph.</p> <h1 style="font-family:verdana;">This is a heading</h1> <p style="font-family:courier;">This is a paragraph.</p> <h1 style="font-size:200%;">This is a heading</h1> <p style="font-size:100%;">This is a paragraph.</p> <h1 style="text-align:center;">Centered Heading</h1> <p style="text-align:center;">Centered paragraph.</p> </body> </html></pre>	<p>This is a heading</p> <p>This is a paragraph.</p> <p>This is a heading</p> <p>This is a paragraph.</p> <p>This is a heading</p> <p>This is a paragraph.</p> <p>Centered Heading</p> <p>Centered paragraph.</p>

4. HTML Text Formatting

- HTML chứa nhiều phần tử để xác định văn bản có ý nghĩa đặc biệt. Ví dụ:

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>in đậm</p> <p><i>in nghiêng</i></p> <p><sub>dưới</sub> và <sup>trên</sup></p> </body> </html></pre>	<p>in đậm</p> <p><i>in nghiêng</i></p> <p>dưới và trên</p>

- Các phần tử định dạng trong HTML được thiết kế để hiển thị các loại văn bản đặc biệt:

Phần tử	Mô tả
	Văn bản in đậm
	Văn bản quan trọng, thường được hiển thị in đậm và có ý nghĩa semantical.
<i>	Văn bản nghiêng.
	Văn bản được nhấn mạnh, thường được hiển thị nghiêng và có ý nghĩa semantical.
<mark>	Văn bản được đánh dấu.
<small>	Văn bản nhỏ hơn.
	Văn bản bị xóa.
<ins>	Văn bản được chèn thêm.
<sub>	Văn bản dưới dòng (subscript).
<sup>	Văn bản trên dòng (superscript).

5. HTML Quotation and Citation Elements

- Phần tử HTML <blockquote> xác định một phần được trích dẫn từ một nguồn khác. Trình duyệt thường thụt đầu dòng cho các phần tử <blockquote>.

Thực tập Kiến trúc và Giao thức IoT

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>Sau đây là quote:</p> <blockquote cite="https://vi.wikipedia.org/wiki/H%E1%BB%93_Ch%C3%AD_Minh"> Ngoài hoạt động chính trị, Hồ Chí Minh cũng là nhà văn, nhà thơ và nhà báo với nhiều tác phẩm viết bằng tiếng Việt, tiếng Hán và tiếng Pháp. </blockquote> </body> </html></pre>	<p>Sau đây là quote:</p> <p>Ngoài hoạt động chính trị, Hồ Chí Minh cũng là nhà văn, nhà thơ và nhà báo với nhiều tác phẩm viết bằng tiếng Việt, tiếng Hán và tiếng Pháp.</p>

- Phần tử HTML `<q>` xác định một trích dẫn ngắn. Trình duyệt thường tự động thêm dấu ngoặc kép xung quanh trích dẫn.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>Bác Hồ đã nói: <q>Không có gì quý hơn độc lập tự do.</q></p> </body> </html></pre>	<p>Bác Hồ đã nói: “Không có gì quý hơn độc lập tự do.”</p>

- Phần tử HTML `<abbr>` xác định một từ viết tắt hoặc viết tắt viết hoa, như "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

+ Đánh dấu các từ viết tắt có thể cung cấp thông tin hữu ích cho trình duyệt, hệ thống dịch và các công cụ tìm kiếm.

+ Sử dụng thuộc tính `title` toàn cục để hiển thị mô tả cho từ viết tắt/ký hiệu khi bạn di chuột qua phần tử.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</pre>	<p>The WHO was founded in 1948.</p> <p>Đánh dấu các từ viết tắt có thể cung cấp thông tin hữu ích cho trình duyệt, hệ thống dịch và các công cụ tìm kiếm.</p>

Thực tập Kiến trúc và Giao thức IoT

```
Organization">WHO</abbr>  
was founded in 1948.</p>  
  
<p>Đánh dấu các từ viết  
tắt có thể cung cấp  
thông tin hữu ích cho  
trình duyệt, hệ thống  
dịch và các công cụ tìm  
kiếm.</p>  
  
</body>  
</html>
```

- Phần tử HTML `<address>` xác định thông tin liên hệ cho tác giả/chủ sở hữu của một tài liệu hoặc một bài viết.

+ Thông tin liên hệ có thể là địa chỉ email, URL, địa chỉ vật lý, số điện thoại, tên tài khoản trên mạng xã hội, v.v.

+ Văn bản trong phần tử `<address>` thường được hiển thị nghiêng, và trình duyệt sẽ luôn thêm một dòng xuống dòng trước và sau phần tử `<address>`.

Code	Result
<pre><!DOCTYPE html> <html> <body> <address> Written by John Doe.
 Visit us at:
 Example.com
 Box 564, Disneyland
 USA </address> </body> </html></pre>	<p><i>Written by John Doe.</i></p> <p><i>Visit us at:</i></p> <p><i>Example.com</i></p> <p><i>Box 564, Disneyland</i></p> <p><i>USA</i></p>

- Phần tử HTML `<cite>` xác định tiêu đề của một tác phẩm sáng tạo (ví dụ: một cuốn sách, một bài thơ, một bài hát, một bộ phim, một bức tranh, một tác phẩm điêu khắc, v.v.).

+ Lưu ý: Tên của một người không phải là tiêu đề của một tác phẩm.

+ Văn bản trong phần tử `<cite>` thường được hiển thị nghiêng.

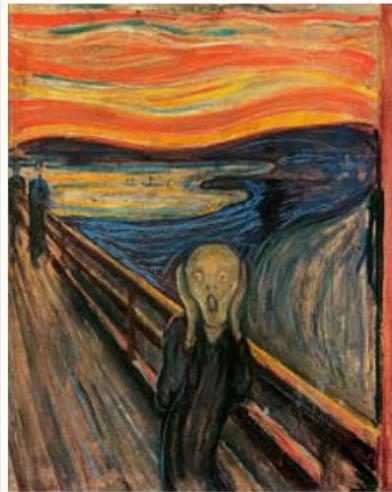
Code	Result
------	--------

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<body>


<p><cite>The Scream</cite> by
Edvard Munch. Painted in
1893.</p>

</body>
</html>
```



The Scream by Edvard Munch. Painted in 1893.

- Phân tử HTML **<bdo>** được sử dụng để ghi đè hướng văn bản hiện tại:

Code	Result
<!DOCTYPE html> <html> <body> <p>Nếu trình duyệt của bạn hỗ trợ ghi đè hướng song hướng (bdo), dòng tiếp theo sẽ được viết từ phải sang trái (rtl):</p> <bdo dir="rtl">Dòng này nè</bdo> </body> </html>	Nếu trình duyệt của bạn hỗ trợ ghi đè hướng song hướng (bdo), dòng tiếp theo sẽ được viết từ phải sang trái (rtl): nè Dòng này

6. HTML Comments

- Bạn có thể thêm các ghi chú vào mã nguồn HTML bằng cách sử dụng cú pháp sau:

```
<!-- Write your comments here -->
```

- Ghi chú: Các ghi chú không được trình duyệt hiển thị, nhưng chúng có thể giúp tài liệu hóa mã nguồn HTML của bạn.

- Bằng cách sử dụng ghi chú, bạn có thể đặt các thông báo và lời nhắc trong mã nguồn HTML của bạn:

Code	Result
<!DOCTYPE html> <html> <body> <!-- Viết comment ở đây nè --> <p>Đoạn văn nè.</p>	Đoạn văn nè.

Thực tập Kiến trúc và Giao thức IoT

```
<!-- Comment không có được hiển thị  
đâu. -->  
</body>  
</html>
```

- Ghi chú có thể được sử dụng để ẩn nội dung.
- Bạn cũng có thể ẩn nhiều dòng hơn. Mọi thứ nằm giữa <!-- và --> sẽ bị ẩn khỏi hiển thị.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>This is a paragraph.</p> <!-- <p>Nhìn nè:</p> --> <p>This is a paragraph too.</p> </body> </html></pre>	<p>This is a paragraph.</p> <p>This is a paragraph too.</p>

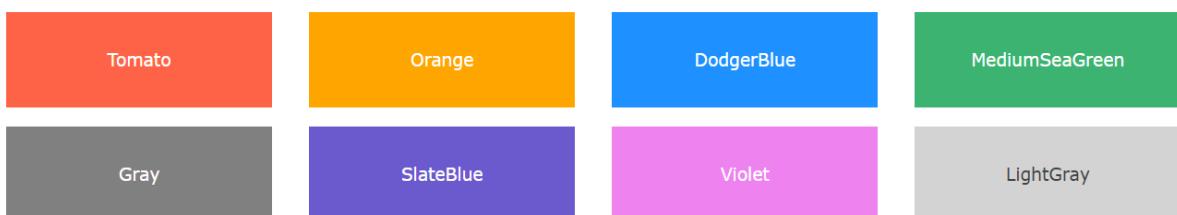
- Ghi chú có thể được sử dụng để ẩn các phần ở giữa mã nguồn HTML.

7. HTML Colors

7.1. Colors

- Màu sắc trong HTML có thể được chỉ định bằng cách sử dụng tên màu được xác định trước, hoặc bằng các giá trị RGB, HEX, HSL, RGBA, hoặc HSLA.

- Trong HTML, có những tên màu như sau:



Hình 13: Tên màu sử dụng trong HTML

- HTML hỗ trợ 140 tên màu chuẩn (standard color names).
- Có thể đặt màu văn bản, màu nền và màu khung viền bằng ‘style’.
- Trong HTML, bạn cũng có thể chỉ định màu sắc bằng cách sử dụng giá trị RGB, HEX, HSL, RGBA và HSLA. Các phần tử <div> sau đây có màu nền được đặt bằng các giá trị RGB, HEX và HSL:

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

- Các phần tử <div> sau đây có màu nền được đặt bằng các giá trị RGBA và HSLA, giúp thêm kênh Alpha (độ trong suốt) vào màu sắc (ở đây chúng ta có độ trong suốt 50%):

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

Code	Result
<!DOCTYPE html>	Giống như tên màu "Tomato":
<html>	<code>rgb(255, 99, 71)</code>
<body>	<code>#ff6347</code>
<p>Giống như tên màu "Tomato":</p>	<code>hsl(9, 100%, 64%)</code>
<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>	Giống như tên màu "Tomato", nhưng có độ trong suốt 50%:
<h1 style="background-color:#ff6347;">#ff6347</h1>	<code>rgba(255, 99, 71, 0.5)</code>
<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>	<code>hsla(9, 100%, 64%, 0.5)</code>
<p>Giống như tên màu "Tomato", nhưng có độ trong suốt 50%:</p>	
<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h1>	
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>	
</body>	
</html>	

7.2. RGB

- Giá trị màu RGB đại diện cho các nguồn sáng ĐỎ (RED), LỤC (GREEN) và LAM (BLUE).

- Giá trị màu RGBA là một sự mở rộng của RGB với một kênh Alpha (độ trong suốt).
- Trong HTML, màu sắc có thể được chỉ định bằng giá trị RGB, sử dụng công thức sau: `rgb(red, green, blue)`

Thực tập Kiến trúc và Giao thức IoT

- Mỗi tham số (đỏ, lục và lam) xác định độ mạnh của màu sắc với giá trị nằm trong khoảng từ 0 đến 255.

- Điều này có nghĩa có tổng cộng $256 \times 256 \times 256 = 16,777,216$ màu sắc có thể tạo ra!

- Ví dụ, `rgb(255, 0, 0)` được hiển thị là màu đỏ, vì màu đỏ được đặt ở giá trị cao nhất (255), và hai màu khác (xanh lục và lam) được đặt thành 0.

- Một ví dụ khác, `rgb(0, 255, 0)` được hiển thị là màu xanh lục, vì màu xanh lục được đặt ở giá trị cao nhất (255), và hai màu khác (đỏ và lam) được đặt thành 0.

- Để hiển thị màu đen, bạn đặt tất cả các tham số màu thành 0, như sau: `rgb(0, 0, 0)`.

- Để hiển thị màu trắng, bạn đặt tất cả các tham số màu thành 255, như sau: `rgb(255, 255, 255)`.

- Các tông màu xám thường được định nghĩa bằng cách sử dụng các giá trị bằng nhau cho cả ba tham số.

- Giá trị màu RGBA là một sự mở rộng của giá trị màu RGB với một kênh Alpha, điều này xác định độ trong suốt cho một màu sắc. Giá trị màu RGBA được chỉ định với cú pháp sau: `rgba(red, green, blue, alpha)`

Code	Result
<!DOCTYPE html> <html> <body> <h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1> <h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1> <h1 style="background-color:rgb(60, 60, 60);">rgb(0, 0, 255)</h1> <h1 style="background-color:rgba(0,0,255,0.5);">rgba(0,0,255,0.5)</h1> </body> </html>	rgb(255, 0, 0) rgb(0, 0, 255) rgb(0, 0, 255) rgba(0,0,255,0.5)

7.3. HEX

- Trong HTML, một màu sắc có thể được chỉ định bằng giá trị thập lục phân (hexadecimal) theo dạng sau: `#rrrggbb`

- Trong đó:

+ # là dấu thăng (#) đặt ở đầu giá trị màu để chỉ ra rằng đó là một giá trị màu sắc.

+ RR (đỏ), GG (lục) và BB (lam) là các số thập lục phân (0-9 và A-F) xác định các thành phần màu đỏ, lục và lam của màu sắc.

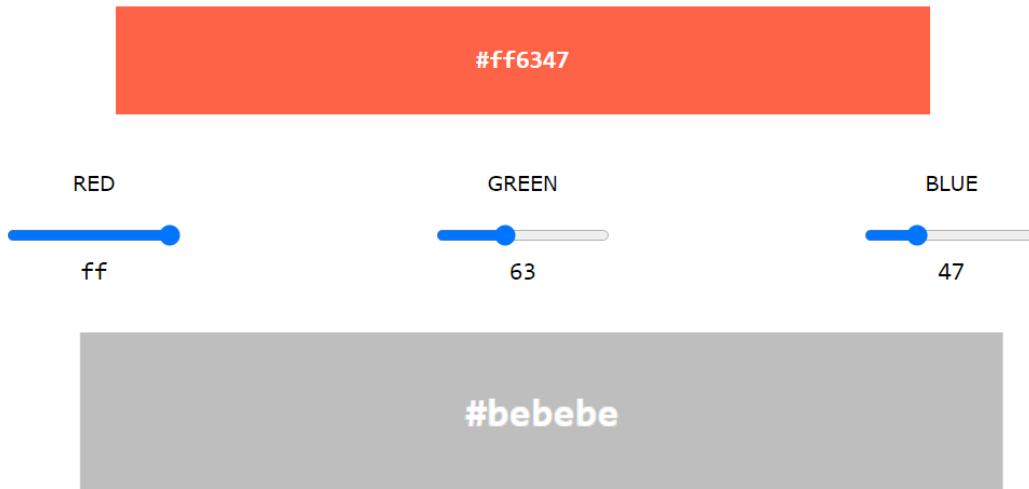
- Ví dụ:

+ `#FF0000` tương ứng với màu đỏ đậm.

+ `#00FF00` tương ứng với màu xanh lục đậm.

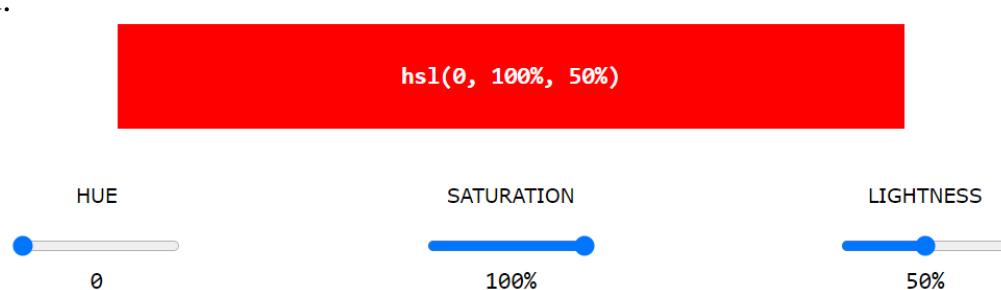
Thực tập Kiến trúc và Giao thức IoT

- + #0000FF tương ứng với màu xanh lam đậm.
- + #000000 tương ứng với màu đen.
- + #ffffff tương ứng với màu trắng.
- Các tông màu xám thường được định nghĩa bằng cách sử dụng các giá trị bằng nhau cho cả ba tham số.
- Ví dụ:



7.4. HSL

- Trong HTML, màu sắc có thể được chỉ định bằng giá trị màu HSL (hue, saturation, và lightness) theo dạng sau: hsl(hue, saturation, lightness)
- Hue là một giá trị độ trên bánh xe màu từ 0 đến 360. 0 tương ứng với màu đỏ, 120 tương ứng với màu xanh lục, và 240 tương ứng với màu xanh lam.
- Saturation là một giá trị phần trăm. 0% đại diện cho các tông màu xám, và 100% đại diện cho màu sắc đầy đủ.
- Lightness cũng là một giá trị phần trăm. 0% đại diện cho màu đen, và 100% đại diện cho màu trắng.
- Ví dụ:



- Độ bão hòa (saturation) có thể mô tả là sự cường độ của màu sắc.
 - + 100% đại diện cho màu sắc nguyên bản, không có các tông màu xám.
 - + 50% đại diện cho một màu sắc kèm theo 50% xám, nhưng bạn vẫn có thể thấy được màu sắc gốc.
 - + 0% đại diện cho màu hoàn toàn xám; bạn không còn thấy được màu sắc gốc.

`hsl(0, 40%, 50%)`

- Độ sáng (lightness) của màu sắc có thể được mô tả bằng cách xác định mức độ ánh sáng bạn muốn cho màu sắc, trong đó:

`hsl(0, 100%, 25%)`

- Các tông màu xám thường được định nghĩa bằng cách đặt giá trị màu (hue) và độ bão hòa (saturation) thành 0 và điều chỉnh độ sáng (lightness) từ 0% đến 100% để có được các tông màu xám đậm hoặc nhạt hơn.

`hsl(0, 0%, 60%)`

- Giá trị màu HSLA là một sự mở rộng của giá trị màu HSL với một kênh Alpha, điều này xác định độ trong suốt cho một màu sắc. Giá trị màu HSLA được chỉ định với cú pháp sau: `hsla(hue, saturation, lightness, alpha)`

- Tham số alpha là một số nằm trong khoảng từ 0.0 (hoàn toàn trong suốt) đến 1.0 (không trong suốt).

`hsla(0, 100%, 50%, 0.5)`



8. HTML CSS

8.1. Using CSS

- CSS có thể được thêm vào tài liệu HTML bằng 3 cách:

- + Inline - bằng cách sử dụng thuộc tính style bên trong các phần tử HTML.

- + Internal - bằng cách sử dụng phần tử `<style>` trong phần `<head>` của tài liệu.

- + External - bằng cách sử dụng phần tử `<link>` để liên kết đến một tệp CSS ngoài.

- Cách phổ biến nhất để thêm CSS là giữ các kiểu trong các tệp CSS ngoài. Tuy nhiên, trong hướng dẫn này, chúng ta sẽ sử dụng kiểu inline và kiểu internal, vì điều này dễ dàng để minh họa và bạn có thể thử nghiệm một cách dễ dàng.

8.2. Inline CSS

- CSS inline được sử dụng để áp dụng kiểu duy nhất cho một phần tử HTML cụ thể.

- CSS inline sử dụng thuộc tính style của một phần tử HTML.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1 style="color:blue;">A Blue Heading</h1> <p style="color:red;">A red paragraph.</p> </body> </html></pre>	<h1>A Blue Heading</h1> <p>A red paragraph.</p>

8.3. Internal CSS

- CSS nội bộ (internal CSS) được sử dụng để định nghĩa kiểu cho một trang HTML cụ thể.

- CSS nội bộ được định nghĩa trong phần `<head>` của một trang HTML, bên trong một phần tử `<style>`.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body {background-color: powderblue;} h1 {color: blue;} p {color: red;} </style> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> </body> </html></pre>	<h1>This is a heading</h1> <p>This is a paragraph.</p>

8.4. External CSS

- Một tệp kiểu CSS ngoài (external style sheet) được sử dụng để định nghĩa kiểu cho nhiều trang HTML.

- Để sử dụng một tệp kiểu CSS ngoài, bạn cần thêm một liên kết đến nó trong phần `<head>` của từng trang HTML:

Code	Result
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" href="styles.css"> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> </body> </html> body { background-color: powderblue; } h1 { color: blue; } p { color: red; }</pre>	<p>This is a heading</p> <p>This is a paragraph.</p>

- Tệp kiểu CSS ngoài có thể được viết trong bất kỳ trình soạn thảo văn bản nào. Tệp này không nên chứa bất kỳ mã HTML nào và phải được lưu với phần mở rộng .css.

8.5. CSS Colors, Fonts and Sizes

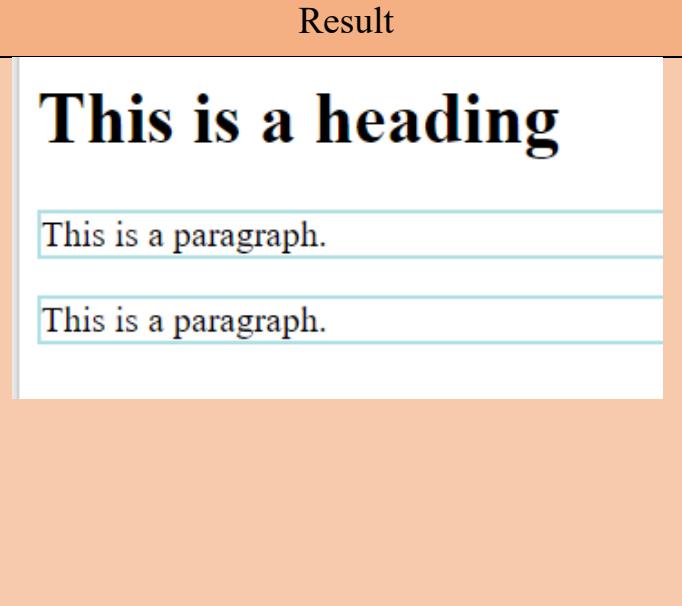
- Một số thuộc tính CSS thường được sử dụng.
 - + Thuộc tính CSS color xác định màu văn bản sẽ được sử dụng.
 - + Thuộc tính CSS font-family xác định phông chữ sẽ được sử dụng.
 - + Thuộc tính CSS font-size xác định kích thước văn bản sẽ được sử dụng.
- Các thuộc tính này giúp điều chỉnh ngoại hình của văn bản trên trang web của bạn.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { color: blue; font-family: verdana; font-size: 300%; } p { color: red; font-family: courier; font-size: 160%; } </style></pre>	<p>This is a heading</p> <p>This is a paragraph.</p>

```
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

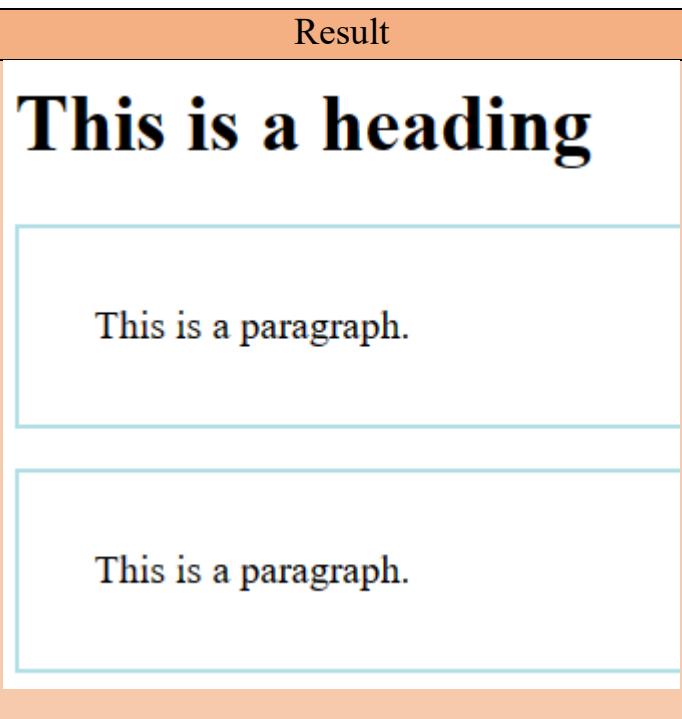
8.6. CSS Border

- Thuộc tính CSS border xác định một viền xung quanh một phần tử HTML.
- Lưu ý: Bạn có thể xác định một viền cho gần như tất cả các phần tử HTML.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { border: 2px solid powderblue; } </style> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> <p>This is a paragraph.</p> </body> </html></pre>	

8.7. CSS Padding

- Thuộc tính CSS ‘padding’ xác định khoảng cách (khoảng trống) giữa văn bản và viền.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { border: 2px solid powderblue; padding: 30px; } </style> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> <p>This is a paragraph.</p> </body> </html></pre>	

8.8. CSS Margin

- Thuộc tính CSS margin xác định một lề (khoảng trống) ở bên ngoài viền của một phần tử.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { border: 2px solid powderblue; margin: 50px; } </style> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> <p>This is a paragraph.</p> </body> </html></pre>	<p>This is a heading</p> <p>This is a paragraph.</p> <p>This is a paragraph.</p>

8.9. Link to External CSS

- Tập kiểu CSS ngoài có thể được tham chiếu bằng một URL đầy đủ hoặc bằng một đường dẫn tương đối đến trang web hiện tại.

Code	Result
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" href="https://www.w3schools.com/htm l/styles.css"> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> </body> </html></pre>	<p>This is a heading</p> <p>This is a paragraph.</p>

9. HTML Images

9.1. Image

a. HTML Images Syntax

- Cú pháp: ``
- Thẻ `` được dùng để nhúng hình ảnh vào trang web, yêu cầu hai thuộc tính:
 - + Src: đường dẫn đến hình ảnh

Thực tập Kiến trúc và Giao thức IoT

+ Alt: Văn bản thay thế cho hình ảnh

b, The src Attribute

- Thuộc tính src được dùng để chèn đường link của hình ảnh cho trang web.

- Cú pháp:

c, The alt Attribute

- Thuộc tính alt được dùng để tạo một text thay thế cho hình ảnh nếu người dùng vì một số lí do nào đó không thể xem được ảnh.

- Cú pháp:

d, Image Size - Width and Height

- Có thể thiết lập kích thước hình ảnh khi chèn vào trang web tùy thuộc vào quyết định của người lập trình để tạo một trang web trông đẹp mắt hơn.

- Kích thước ảnh không phù hợp với kích thước trang web do đó có thể thay đổi giá trị của width và height để phù hợp hơn.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Image Size</h2> <p>Here we use the style attribute to specify the width and height of an image:</p> </body> </html></pre>	<p>Image Size</p> <p>Here we use the style attribute to specify the width and height of an image:</p> 

f, Width and Height, or Style?

- Có thể sử dụng width và height hoặc style để điều chỉnh kích thước hình ảnh, nhưng tốt hơn nên sử dụng style để hạn chế style sheets tự thay đổi kích thước của hình ảnh.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> img { width: 100%; } </style> </head></pre>	

Thực tập Kiến trúc và Giao thức IoT

```
<body>





</body>
</html>
```



g, Images in Another Folder

- Nếu chèn ảnh từ thư mục khác, cần dẫn đường dẫn đến thư mục trong thuộc tính src.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Images in Another Folder</h2> <p>It is common to store images in a sub-folder. You must then include the folder name in the src attribute:</p> </body> </html></pre>	<p>Images in Another Folder</p> <p>It is common to store images in a sub-folder. You must then include the folder name in the src attribute:</p> 

h, Images on Another Server/Website

- HTML cũng hỗ trợ chèn ảnh từ trang web khác, cũng như hỗ trợ ảnh động (GIFs).

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Images on Another Server</h2> </body> </html></pre>	<p>Images on Another Server</p> 

Thực tập Kiến trúc và Giao thức IoT

i, Image as a Link

- Img cũng có thể được dùng làm nội dung trong thẻ `<a>` để khi ấn vào ảnh sẽ dẫn đến một đường liên kết mới.
- Sau khi nhấp vào icon trên hình giúp chuyển hướng đến một trang web mới.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Image as a Link</h2> <p>The image is a link. You can click on it.</p> </body> </html></pre>	<p>Image as a Link</p> <p>The image is a link. You can click on it.</p> 

j, Image Floating

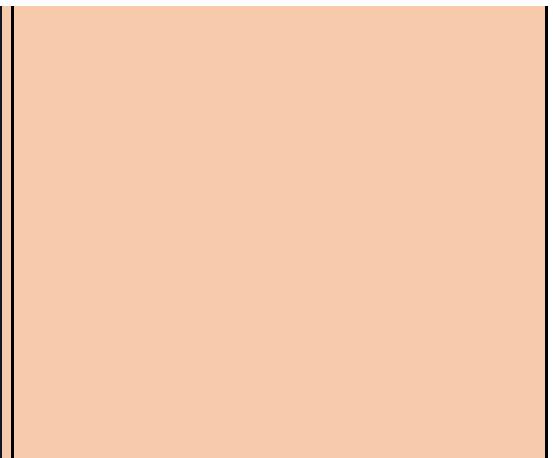
- Có thể sử dụng thuộc tính float của CSS để thiết lập vị trí của hình ảnh trong trang web của bạn.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Floating Images</h2> <p>Float the image to the right:</p> <p> </p> A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image. </p></pre>	<p>Floating Images</p> <p>Float the image to the right:</p> <p>A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.</p> <p>Float the image to the left:</p> <p> A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.</p>

```
<p><strong>Float the image to the left:</strong></p>
<p>

A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.
</p>

</body>
</html>
```



9.2. Image Maps

- Thủ `<map>` định nghĩa một image map. Là một ảnh với khu vực có thể được click vào. Vùng đó được định nghĩa với 1 hoặc nhiều thẻ `<area>`.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Image Maps</h2> <p>Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:</p> <map name="workmap"> <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm"> <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm"> <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm"> </map> </body> </html></pre>	<p>Image Maps</p> <p>Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:</p>

- Khi click vào computer hoặc coffee cup trong image trên, trang web tự động chuyển hướng đến đường link đã dẫn trước đó.

Thực tập Kiến trúc và Giao thức IoT

a, The Image

- The image được chèn bằng cách sử dụng thẻ ``. Điểm khác nhau là bạn phải sử dụng thuộc tính `usemap`.

- Cú pháp: ``

b, Create Image Map

- Thẻ `<map>` được sử dụng để tạo một image map. Và được liên kết với image bằng cách sử dụng thuộc tính `name`.

- Cú pháp: `<map name="workmap">`

c, The Areas

- Khoảng có thể click vào được định nghĩa bằng phần tử `<area>`.

d, Shape

- Thuộc tính `shape` định nghĩa hình dạng của vùng có thể click vào. Có thể chọn một trong số các giá trị sau:

+ Rect – định nghĩa một vùng hình hộp chữ nhật:

`<area shape="rect" coords="34, 44, 270, 350" href="computer.htm">`

+ Circle – định nghĩa một vùng hình tròn:

`<area shape="circle" coords="337, 300, 44" href="coffee.htm">`

+ Poly – định nghĩa một vùng hình đa giác:

`<area shape="poly"`

`coords="140,121,181,116,204,160,204,222,191,270,140,329,85,355,58,352,37,32`
`2,40,259,103,161,128,147" href="croissant.htm">`

+ Default – định nghĩa toàn bộ hình ảnh.

e, Image Map and JavaScript

- Một vùng có thể click vào cũng cần một trigger từ hàm Javascript

- Tạo sự kiện click cho phần tử `<area>` để thực thi hàm Javascript

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Image Maps</h2> <p>Click on the cup of coffee to execute a JavaScript function:</p> <map name="workmap"></pre>	<p>Image Maps</p> <p>Click on the cup of coffee to execute a JavaScript function:</p> 

```

<area shape="circle"
coords="337,300,44" href="coffee.htm"
onclick="myFunction()">
</map>

<script>
function myFunction() {
  alert("You clicked the coffee cup!");
}
</script>

</body>
</html>

```

www.w3schools.com says

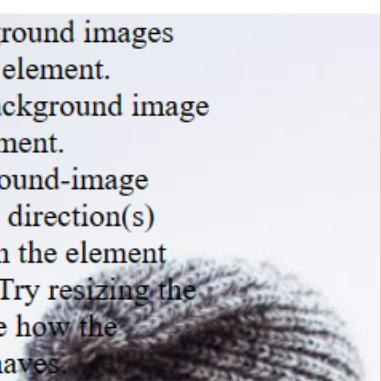
You clicked the coffee cup!

OK

9.3. Background Image

a, Background Image on a HTML element

- Để tạo một background image cho một phần tử HTML, có thể dùng thuộc tính style và CSS background-image property.

Code	Result
<pre> <!DOCTYPE html> <html> <body> <h2>Background Image</h2> <p>A background image for a p element:</p> <p style="background-image: url('img_girl.jpg');"> You can specify background images
 for any visible HTML element.
 In this example, the background image
 is specified for a p element.
 By default, the background- image
 will repeat itself in the direction(s)
 where it is smaller than the element
 where it is specified. (Try resizing the
 browser window to see how the
 background image behaves. </p> </body> </html> </pre>	<p>Background Image</p> <p>A background image for a p element:</p> <p>You can specify background images for any visible HTML element. In this example, the background image is specified for a p element. By default, the background-image will repeat itself in the direction(s) where it is smaller than the element where it is specified. (Try resizing the browser window to see how the background image behaves.)</p> 

Thực tập Kiến trúc và Giao thức IoT

b, Background Image on a Page

- Nếu bạn muốn tạo toàn bộ trang web của bạn một hình nền, bạn có thể chỉ định background image trong thẻ <body>.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url('img_girl.jpg'); } </style> </head> <body> <h2>Background Image</h2> <p>By default, the background image will repeat itself if it is smaller than the element where it is specified, in this case the body element.</p> </body> </html></pre>	<p>Background Image</p> <p>By default, the background image will repeat itself if it is smaller than the element where it is specified, in this case the body element.</p> 

c, Background Repeat

- Nếu hình nền nhỏ hơn so với phần tử, nó sẽ tự động lặp lại, cho đến khi nào chạm đến điểm cuối của phần tử.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url('example_img_girl.jpg'); } </style> </head> <body> <h2>Background Repeat</h2> <p>By default, the background image will repeat itself if it is smaller than the element where it is</pre>	<p>Background Repeat</p> <p>By default, the background image will repeat itself if it is smaller than the element where it is specified, in this case the body element.</p> 

```
specified, in this case the body  
element.</p>  
  
</body>  
</html>
```

- Để tránh hiện tượng này có thể đặt background-repeat property ở chế độ no-repeat.

d, Background Cover

- Nếu bạn muốn hình nền của bạn bao phủ lấy toàn bộ phần tử, bạn có thể đặc chế độ background-size ở chế độ cover.

- Để chắc chắn rằng toàn bộ phần tử được bao phủ, đặt background-attachment property ở chế độ fixed.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url('img_girl.jpg'); background-repeat: no-repeat; background-attachment: fixed; background-size: cover; } </style> </head> <body> <h2>Background Cover</h2> <p>Set the background-size property to "cover" and the background image will cover the entire element, in this case the body element.</p> </body> </html></pre>	<p>Background Cover</p> <p>Set the background-size property to "cover" and the background image will cover the entire element, in this case the body element.</p> 

e, Background Stretch

- Nếu bạn muốn hình ảnh kéo dài để vừa với toàn bộ phần tử, bạn có thể đặt background-size property ở mức 100%.

Code	Result
<pre data-bbox="165 233 727 1163"><!DOCTYPE html> <html> <head> <style> body { background-image: url('img_girl.jpg'); background-repeat: no-repeat; background-attachment: fixed; background-size: 100% 100%; } </style> </head> <body> <h2>Background Stretch</h2> <p>Set the background-size property to "100% 100%" and the background image will be stretched to cover the entire element, in this case the body element.</p> </body> </html></pre>	<p data-bbox="854 233 1038 258">Background Stretch</p> <p data-bbox="854 271 1435 309">Set the background-size property to "100% 100%" and the background image will be stretched to cover the entire element, in this case the body element.</p> 

9.4. The Picture Element

- Thẻ <picture> được dùng để chấp nhận để hiển thị các bức ảnh khác nhau trên các thiết bị khác nhau.
- Thẻ <picture> cho phép người lập trình linh hoạt hơn trong việc chỉ định các hình ảnh.
- Thẻ <picture> bao gồm một hoặc nhiều thẻ <source>, mỗi thẻ chỉ định một hình ảnh khác nhau thông qua thuộc tính srcset. Ở cách này trình duyệt có thể chọn hình ảnh phù hợp nhất với thiết bị hiện tại.

Code	Result
<pre data-bbox="165 1691 822 2061"><!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width, initial-scale=1.0"> </head> <body> <h2>The picture Element</h2></pre>	<p data-bbox="886 1691 1049 1715">The picture Element</p>  <p data-bbox="886 1963 1419 2001">Resize the browser to see different versions of the picture loading at different viewport sizes. The browser looks for the first source element where the media query matches the user's current viewport width, and fetches the image specified in the srcset attribute.</p> <p data-bbox="886 2010 1410 2048">The img element is required as the last child tag of the picture declaration block. The img element is used to provide backward compatibility for browsers that do not support the picture element, or if none of the source tags matched.</p>

```
<picture>
  <source media="(min-width: 650px)"
srcset="img_food.jpg">
  <source media="(min-width: 465px)"
srcset="img_car.jpg">
  
</picture>

<p>Resize the browser to see different
versions of the picture loading at
different viewport sizes.
The browser looks for the first source
element where the media query matches
the user's current viewport width,
and fetches the image specified in the
srcset attribute.</p>

<p>The img element is required as the
last child tag of the picture
declaration block.
The img element is used to provide
backward compatibility for browsers
that do not support the picture
element, or if none of the source tags
matched.
</p>

</body>
</html>
```

- Khi nào sẽ dùng thẻ <picture>:

+ Bandwidth: Nếu bạn có một thiết bị với màn hình nhỏ, không cần thiết để tải một ảnh lớn, do đó trình duyệt sẽ chọn <source> đầu tiên với thuộc tính phù hợp và loại bỏ các source còn lại.

+ Format Support: Một số trình duyệt không hỗ trợ toàn bộ format của hình ảnh. Bằng việc sử dụng phần tử picture, bạn có thể thêm các hình ảnh với tất cả formats, trình duyệt sẽ sử dụng formats đầu tiên mà nó nhận ra và từ chối các format còn lại.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width, initial- scale=1.0"> </head> <body> <h2>The picture Element</h2> <picture> <source srcset="img_avatar.png"> <source srcset="img_girl.jpg"> </picture> <p>The picture element can be used when the image format is not supported by all devices.</p> <p>The device will use the first image format it supports, and ignore the rest of the images.</p> </body> </html> </pre>	<p>The picture Element</p>  <p>The picture element can be used when the image format is not supported by all devices. The device will use the first image format it supports, and ignore the rest of the images.</p>

10. HTML Tables

10.1. HTML Tables

- HTML cung cấp bảng để người phát triển có thể sắp xếp dữ liệu trong các hàng và các cột.

a, Define an HTML Table

- Một bảng HTML bao gồm các ô giữa các hàng và cột.

Code	Result									
<pre> <!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; } </style> <body> </pre>	<p>A basic HTML table</p> <table border="1"> <thead> <tr> <th>Company</th> <th>Contact</th> <th>Country</th> </tr> </thead> <tbody> <tr> <td>Alfreds Futterkiste</td> <td>Maria Anders</td> <td>Germany</td> </tr> <tr> <td>Centro comercial Moctezuma</td> <td>Francisco Chang</td> <td>Mexico</td> </tr> </tbody> </table> <p>To understand the example better, we have added borders to the table.</p>	Company	Contact	Country	Alfreds Futterkiste	Maria Anders	Germany	Centro comercial Moctezuma	Francisco Chang	Mexico
Company	Contact	Country								
Alfreds Futterkiste	Maria Anders	Germany								
Centro comercial Moctezuma	Francisco Chang	Mexico								

```
<h2>A basic HTML table</h2>

<table style="width:100%">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial
Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

<p>To understand the example
better, we have added borders to
the table.</p>

</body>
</html>
```

b, Table Cells

- Mỗi ô trong bảng được định nghĩa bởi cặp thẻ `<td>` và `</td>`;
- Mọi thứ giữa cặp thẻ đó là nội dung của ô trong bảng.

Code	Result			
<pre><!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; } </style> <body> <h2>TD elements define table cells</h2> <table style="width:100%"> <tr> <td>Emil</td></pre>	<p>TD elements define table cells</p> <table border="1"><tr><td>Emil</td><td>Tobias</td><td>Linus</td></tr></table> <p>To understand the example better, we have added borders to the table.</p>	Emil	Tobias	Linus
Emil	Tobias	Linus		

Thực tập Kiến trúc và Giao thức IoT

```
<td>Tobias</td>
<td>Linus</td>
</tr>
</table>

<p>To understand the example better, we have added borders to the table.</p>

</body>
</html>
```

c, Table Rows

- Mỗi hàng trong bảng được bắt đầu với thẻ `<tr>` và kết thúc với thẻ `</tr>`.

Code	Result						
<pre><!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; } </style> <body> <h2>TR elements define table rows</h2> <table style="width:100%"> <tr> <td>Emil</td> <td>Tobias</td> <td>Linus</td> </tr> <tr> <td>16</td> <td>14</td> <td>10</td> </tr> </table> <p>To understand the example better, we have added borders to the table.</p> </body> </html></pre>	<p>TR elements define table rows</p> <table border="1"><tr><td>Emil</td><td>Tobias</td><td>Linus</td></tr><tr><td>16</td><td>14</td><td>10</td></tr></table> <p>To understand the example better, we have added borders to the table.</p>	Emil	Tobias	Linus	16	14	10
Emil	Tobias	Linus					
16	14	10					

d, Table Headers

- Tiêu đề của bảng (tiêu đề của các cột) có thể được thêm vào bảng bằng cặp thẻ `<th></th>`.

Code	Result
------	--------

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
    border:1px solid black;
}
</style>
<body>

<h2>TH elements define table headers</h2>



| Person 1 | Person 2 | Person 3 |
|----------|----------|----------|
| Emil     | Tobias   | Linus    |
| 16       | 14       | 10       |



<p>To understand the example better, we have added borders to the table.</p>

</body>
</html>
```

TH elements define table headers

Person 1	Person 2	Person 3
Emil	Tobias	Linus
16	14	10

To understand the example better, we have added borders to the table.

10.2. HTML Table Border

a, Thêm viền

- Để tạo viền cho bảng, có thể sử dụng CSS border property trên các thẻ table, th và td.

Code

Result

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>

<h2>Table With Border</h2>

<p>Use the CSS border property to add a border to the table.</p>

<table style="width:100%">
<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
</tr>
<tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
</tr>
<tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
</tr>
<tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
</tr>
</table>

</body>
</html>
```

Table With Border

Use the CSS border property to add a border to the table.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

b, Collapsed Table Borders

- Để tránh việc có hai viền cho bảng như ví dụ trên, có thể sử dụng border-collapse property để làm gọn

- Nó sẽ chuyển table trở nên gọn hơn với chỉ một đường viền

Code

Result

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Collapsed Borders</h2>
<p>If you want the borders to collapse into one border, add the CSS border-collapse property.</p>

<table style="width:100%">
    <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
    </tr>
</table>

</body>
</html>
```

Collapsed Borders

If you want the borders to collapse into one border, add the CSS border-collapse property.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

c, Style Table Borders

- Nếu bạn tạo màu nền cho mỗi ô, và đặt border ở màu trắng, có thể tạo ra một border ẩn.

Code	Result
------	--------

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid white;
    border-collapse: collapse;
}
th, td {
    background-color: #96D4D4;
}
</style>
</head>
<body>

<h2>Table With Invisible Borders</h2>

<p>Style the table with white borders  
and a background color of the cells  
to make the impression of invisible  
borders.</p>

<table style="width:100%">
    <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
    </tr>
</table>

</body>
</html>
```

Table With Invisible Borders

Style the table with white borders and a background color of the cells to make the impression of invisible borders.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

d, Round Table Borders

GVHD: Trương Quang Phúc

Thực tập Kiến trúc và Giao thức IoT

- Với border-radius property, bạn có thể tạo được bán kính cho góc của các ô.

Code	Result												
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-radius: 10px; } </style> </head> <body> <h2>Table With Rounded Borders</h2> <p>Use the CSS border-radius property to add rounded corners to the borders.</p> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </table> </body> </html></pre>	<p>Table With Rounded Borders</p> <p>Use the CSS border-radius property to add rounded corners to the borders.</p> <table border="1"><thead><tr><th>Firstname</th><th>Lastname</th><th>Age</th></tr></thead><tbody><tr><td>Jill</td><td>Smith</td><td>50</td></tr><tr><td>Eve</td><td>Jackson</td><td>94</td></tr><tr><td>John</td><td>Doe</td><td>80</td></tr></tbody></table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

e, Dotted Table Borders

GVHD: Trương Quang Phúc

Thực tập Kiến trúc và Giao thức IoT

- Với CSS border-style property, bạn có thể đặt chế độ hiển thị của đường viền.

Code	Result												
<pre><!DOCTYPE html> <html> <head> <style> th, td { border-style: dotted; } </style> </head> <body> <h2>Table With Dotted Borders</h2> <p>Use the CSS border-style property to set the style of the borders.</p> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </table> </body> </html></pre>	<p>Table With Dotted Borders</p> <p>Use the CSS border-style property to set the style of the borders.</p> <table border="1"><thead><tr><th>Firstname</th><th>Lastname</th><th>Age</th></tr></thead><tbody><tr><td>Jill</td><td>Smith</td><td>50</td></tr><tr><td>Eve</td><td>Jackson</td><td>94</td></tr><tr><td>John</td><td>Doe</td><td>80</td></tr></tbody></table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

- Các giá trị mà property này chấp nhận:

- `dotted`
 - `dashed`
 - `solid`
 - `double`
 - `groove`
 - `ridge`
 - `inset`
 - `outset`
 - `none`
 - `hidden`
- 

f, Border Color

- Với CSS border-color property, bạn có thể cài đặt màu cho đường viền của bạn

Code	Result												
<pre><!DOCTYPE html> <html> <head> <style> th, td { border-style:solid; border-color: #96D4D4; } </style> </head> <body> <h2>Table With Border Color</h2> <p>Use the CSS border-color property to set the color of the borders.</p> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> </table></pre>	<p>Table With Border Color</p> <p>Use the CSS border-color property to set the color of the borders.</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Age</th></tr> </thead> <tbody> <tr> <td>Jill</td><td>Smith</td><td>50</td></tr> <tr> <td>Eve</td><td>Jackson</td><td>94</td></tr> <tr> <td>John</td><td>Doe</td><td>80</td></tr> </tbody> </table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

```

</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
<tr>
  <td>John</td>
  <td>Doe</td>
  <td>80</td>
</tr>
</table>

</body>
</html>

```

10.3. HTML Table Sizes

a, HTML Table Width

Code	Result												
<pre> <!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; border-collapse: collapse; } </style> <body> <h2>100% wide HTML Table</h2> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> </pre>	<p>100% wide HTML Table</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Age</th></tr> </thead> <tbody> <tr> <td>Jill</td><td>Smith</td><td>50</td></tr> <tr> <td>Eve</td><td>Jackson</td><td>94</td></tr> <tr> <td>John</td><td>Doe</td><td>80</td></tr> </tbody> </table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

```

<tr>
  <td>John</td>
  <td>Doe</td>
  <td>80</td>
</tr>
</table>

</body>
</html>

```

b, HTML Table Column Width

- Để tạo kích cỡ cho một cột nhất định, thêm thuộc tính style vào thẻ `<th>` hoặc `<td>`.

Code	Result												
<pre> <!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; border-collapse: collapse; } </style> <body> <h2>Set the first column to 70% of the table width</h2> <table style="width:100%"> <tr> <th style="width:70%">Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </table> </pre>	<p>Set the first column to 70% of the table width</p> <table border="1"> <thead> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> </thead> <tbody> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </tbody> </table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

```
</tr>
</table>

</body>
</html>
```

c, HTML Table Row Height

- Để thiết lập chiều cao cho một cột cụ thể, thêm thuộc tính style vào thẻ <tr>.

Code	Result												
<pre><!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; border-collapse: collapse; } </style> <body> <h2>Set the height of the second row to 200 pixels</h2> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr style="height:200px"> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </table> </body> </html></pre>	<p>Set the height of the second row to 200 pixels</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Age</th></tr> </thead> <tbody> <tr> <td>Jill</td><td>Smith</td><td>50</td></tr> <tr> <td>Eve</td><td>Jackson</td><td>94</td></tr> <tr> <td>John</td><td>Doe</td><td>80</td></tr> </tbody> </table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

10.4. HTML Table Headers

a, HTML Table Headers

- Table headers được định nghĩa với thẻ th. Mỗi phần tử th biểu diễn lại một ô của bảng.

Code	Result									
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <h2>Table Headers</h2> <p>Use the TH element to define table headers.</p> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> </table> </body> </html></pre>	<p>Table Headers</p> <p>Use the TH element to define table headers.</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Age</th></tr> </thead> <tbody> <tr> <td>Jill</td><td>Smith</td><td>50</td></tr> <tr> <td>Eve</td><td>Jackson</td><td>94</td></tr> </tbody> </table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94
Firstname	Lastname	Age								
Jill	Smith	50								
Eve	Jackson	94								

b, Vertical Table Headers

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Vertical Table Headers</h2>

<p>The first column becomes
table headers if you set the
first table cell in each table
row to a TH element:</p>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <td>Jill</td>
    <td>Eve</td>
  </tr>
  <tr>
    <th>Lastname</th>
    <td>Smith</td>
    <td>Jackson</td>
  </tr>
  <tr>
    <th>Age</th>
    <td>50</td>
    <td>94</td>
  </tr>
</table>

</body>
</html>
```

Vertical Table Headers

The first column becomes table headers if you set the first table cell in each table row to a TH element:

Firstname	Jill	Eve
Lastname	Smith	Jackson
Age	50	94

c, Align Table Headers

- Mặc định, header của bạn được in đậm và căn giữa.
- Để thay đổi, bạn có thể dùng CSS text-align property.

Code

Result

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th {
    text-align: left;
}
</style>
</head>
<body>

<h2>Left-align Headers</h2>

<p>To left-align the table headers, use the CSS text-align property.</p>

<table style="width:100%">
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>

</body>
</html>
```

Left-align Headers

To left-align the table headers, use the CSS text-align property.

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

d, Header for Multiple Columns

- Bạn có thể có header cho hai hoặc nhiều cột.
- Để làm điều này, có thể sử dụng thuộc tính colspan của thẻ <th>.

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>A header that spans two columns</h2>

<p>Use the colspan attribute to have a header span over multiple columns.</p>

<table style="width:100%">
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
</body>
</html>
```

A header that spans two columns

Use the colspan attribute to have a header span over multiple columns.

	Name	Age
Jill	Smith	50
Eve	Jackson	94

e, Table Caption

Bạn có thể tạo một caption giống như heading cho toàn bộ bảng của bạn bằng cách sử dụng thẻ `<caption>`.

Code

Result

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
  text-align: left;
}
</style>
</head>
<body>

<h2>Table Caption</h2>
<p>To add a caption to a table, use the caption tag.</p>

<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>

</body>
</html>
```

Table Caption

To add a caption to a table, use the `caption` tag.

Monthly savings

Month	Savings
January	\$100
February	\$50

- Chú ý: thẻ `<caption>` nên được chèn thêm ngay sau thẻ `<table>`

10.5. HTML Table Padding & Spacing

a, HTML Table - Cell Padding

- Cell padding là khoảng cách giữa cá đường viền của ô với nội dung của ô.
- Mặc định cell padding là 0.
- Để thêm padding cho các ô của bảng, sử dụng CSS padding property.

Code	Result												
<pre> <!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } th, td { padding: 15px; } </style> </head> <body> <h2>Cellpadding</h2> <p>Cell padding specifies the space between the cell content and its borders.</p> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </table> <p>Tip: Try to change the padding to 5px.</p> </body> </html> </pre>	<p>Cellpadding</p> <p>Cell padding specifies the space between the cell content and its borders.</p> <table border="1"> <thead> <tr> <th data-bbox="747 339 1017 384">Firstname</th><th data-bbox="1017 339 1271 384">Lastname</th><th data-bbox="1271 339 1446 384">Age</th></tr> </thead> <tbody> <tr> <td data-bbox="747 384 1017 473">Jill</td><td data-bbox="1017 384 1271 473">Smith</td><td data-bbox="1271 384 1446 473">50</td></tr> <tr> <td data-bbox="747 473 1017 563">Eve</td><td data-bbox="1017 473 1271 563">Jackson</td><td data-bbox="1271 473 1446 563">94</td></tr> <tr> <td data-bbox="747 563 1017 631">John</td><td data-bbox="1017 563 1271 631">Doe</td><td data-bbox="1271 563 1446 631">80</td></tr> </tbody> </table> <p>Tip: Try to change the padding to 5px.</p>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

- Để thêm padding chỉ phía trên nội dung, cần sử dụng CSS padding-top property. Và các phía còn lại tương tự có các CSS padding-bottom, CSS padding-left và CSS padding-right property.

b, HTML Table - Cell Spacing

- Cell spacing là khoảng cách giữa mỗi ô.
- Mặc định cell spacing là 2 px.
- Để thay đổi giá trị này cần sử dụng CSS border-spacing property trên phần tử <table>.

Code	Result												
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } table { border-spacing: 30px; } </style> </head> <body> <h2>Cellspacing</h2> <p>Change the space between the cells with the border-spacing property.</p> <table style="width:100%"> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </table></pre>	<p>Cellspacing</p> <p>Change the space between the cells with the border-spacing property.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Firstname</th> <th style="text-align: center;">Lastname</th> <th style="text-align: center;">Age</th> </tr> </thead> <tbody> <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> <td>94</td> </tr> <tr> <td>John</td> <td>Doe</td> <td>80</td> </tr> </tbody> </table>	Firstname	Lastname	Age	Jill	Smith	50	Eve	Jackson	94	John	Doe	80
Firstname	Lastname	Age											
Jill	Smith	50											
Eve	Jackson	94											
John	Doe	80											

```

</tr>
<tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
</tr>
</table>

</body>
</html>

```

10.6. HTML Table Colspan & Rowspan

a, HTML Table - Colspan

- Để tạo một ô trải dài trên nhiều dòng, sử dụng thuộc tính colspan.

- Code	Result						
<pre> <!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <h2>Cell that spans two columns</h2> <p>To make a cell span more than one column, use the colspan attribute.</p> <table style="width:100%"> <tr> <th colspan="2">Name</th> <th>Age</th> </tr> <tr> <td>Jill</td> <td>Smith</td> <td>43</td> </tr> <tr> <td>Eve</td> <td>Jackson</td> </tr> </pre>	<p>Cell that spans two columns</p> <p>To make a cell span more than one column, use the colspan attribute.</p> <table border="1"> <thead> <tr> <th style="text-align: center;">Name</th><th style="text-align: center;">Age</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">Jill</td><td style="text-align: center;">Smith</td></tr> <tr> <td style="text-align: center;">Eve</td><td style="text-align: center;">Jackson</td></tr> </tbody> </table>	Name	Age	Jill	Smith	Eve	Jackson
Name	Age						
Jill	Smith						
Eve	Jackson						

```
<td>57</td>
</tr>
</table>
</body>
</html>
```

b, HTML Table – Rowspan

- Để tạo một ô trải dài trên nhiều hàng, sử dụng thuộc tính rowspan

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <h2>Cell that spans two rows</h2> <p>To make a cell span more than one row, use the rowspan attribute.</p> <table style="width:100%"> <tr> <th>Name</th> <td>Jill</td> </tr> <tr> <th rowspan="2">Phone</th> <td>555-1234</td> </tr> <tr> <td>555-8745</td> </tr> </table> </body> </html></pre>	<p>Cell that spans two rows</p> <p>To make a cell span more than one row, use the rowspan attribute.</p> <table border="1"><thead><tr><th>Name</th><th>Jill</th></tr></thead><tbody><tr><td>Phone</td><td>555-1234</td></tr><tr><td></td><td>555-8745</td></tr></tbody></table>	Name	Jill	Phone	555-1234		555-8745
Name	Jill						
Phone	555-1234						
	555-8745						

10.7. Table Styling

a, HTML Table - Zebra Stripes

Code	Result															
<pre><!DOCTYPE html> <html> <head> <style> table { border-collapse: collapse; width: 100%; } th, td { text-align: left; padding: 8px; } tr:nth-child(even) { background-color: #D6EEEE; } </style> </head> <body> <h2>Zebra Striped Table</h2> <p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p> <table> <tr> <th>First Name</th> <th>Last Name</th> <th>Points</th> </tr> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> <tr> <td>Lois</td> <td>Griffin</td> <td>\$150</td> </tr> <tr></pre>	<p>Zebra Striped Table</p> <p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p> <table> <thead> <tr> <th>First Name</th> <th>Last Name</th> <th>Points</th> </tr> </thead> <tbody> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> <tr> <td>Lois</td> <td>Griffin</td> <td>\$150</td> </tr> <tr> <td>Joe</td> <td>Swanson</td> <td>\$300</td> </tr> <tr> <td>Cleveland</td> <td>Brown</td> <td>\$250</td> </tr> </tbody> </table>	First Name	Last Name	Points	Peter	Griffin	\$100	Lois	Griffin	\$150	Joe	Swanson	\$300	Cleveland	Brown	\$250
First Name	Last Name	Points														
Peter	Griffin	\$100														
Lois	Griffin	\$150														
Joe	Swanson	\$300														
Cleveland	Brown	\$250														

Thực tập Kiến trúc và Giao thức IoT

```
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>

</body>
</html>
```

b, HTML Table - Vertical Zebra Stripes

- Để tạo sọc vằn cho mỗi cột khác nhau, sử dụng câu lệnh như sau:

Code	Result																																			
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } th:nth-child(even),td:nth-child(even) { background-color: #D6EEEE; } </style> </head> <body> <h2>Striped Table</h2> <p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p> <table style="width:100%"> <tr> <th>MON</th> <th>TUE</th> <th>WED</th> <th>THU</th></pre>	<p>Striped Table</p> <p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p> <table border="1"><thead><tr><th>MON</th><th>TUE</th><th>WED</th><th>THU</th><th>FRI</th><th>SAT</th><th>SUN</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>	MON	TUE	WED	THU	FRI	SAT	SUN																												
MON	TUE	WED	THU	FRI	SAT	SUN																														

Thực tập Kiến trúc và Giao thức IoT

```
<th>FRI</th>
<th>SAT</th>
<th>SUN</th>
</tr>
<tr>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
<tr>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
<tr>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
<tr>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
<tr>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
</tr>
</table>

</body>
</html>
```

c, Combine Vertical and Horizontal Zebra Stripes

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}

tr:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}

th:nth-child(even),td:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}
</style>
</head>
<body>

<h2>Striped Table</h2>
<p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p>

<table style="width:100%">
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
    <th>FRI</th>
    <th>SAT</th>
    <th>SUN</th>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
```

Striped Table

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

MON	TUE	WED	THU	FRI	SAT	SUN

```
<td> </td>
</tr>
<tr>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
</tr>
<tr>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
</tr>
</table>

</body>
</html>
```

10.8. Table Colgroup

a, HTML Table Colgroup

- Thủ `<colgroup>` được dùng để tạo style cho các cột đặc biệt của bảng.
- Nếu bạn muốn điều chỉnh trạng thái cho hai cột đầu tiên của bảng, có thể sử dụng thẻ `<colgroup>` và thẻ `<col>`.
 - Mỗi group được chỉ định với một phần tử `<col>`.
 - Thuộc tính `span` chỉ định có bao nhiêu cột được sử dụng để điều chỉnh trạng thái.
 - Thuộc tính `style` chỉ định trạng thái của các cột.

Thực tập Kiến trúc và Giao thức IoT

Code	Result																																			
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <h2>Colgroup</h2> <p>Add the a colgroup with a col element that spans over two columns to define a style for the two columns:</p> <table style="width: 100%;"> <colgroup> <col span="2" style="background-color: #D6EEEE"> </colgroup> <tr> <th>MON</th> <th>TUE</th> <th>WED</th> <th>THU</th> <th>FRI</th> <th>SAT</th> <th>SUN</th> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td></pre>	<p>Colgroup</p> <p>Add the a colgroup with a col element that spans over two columns to define a style for the two columns:</p> <table border="1"><thead><tr><th>MON</th><th>TUE</th><th>WED</th><th>THU</th><th>FRI</th><th>SAT</th><th>SUN</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr><tr><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr><tr><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr></tbody></table>	MON	TUE	WED	THU	FRI	SAT	SUN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
MON	TUE	WED	THU	FRI	SAT	SUN																														
1	2	3	4	5	6	7																														
8	9	10	11	12	13	14																														
15	16	17	18	19	20	21																														
22	23	24	25	26	27	28																														

```

<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
</table>

</body>
</html>

```

b, Legal CSS Properties

- Có một số lượng giới hạn property của CSS cho phép sử dụng trong colgroup.

[width](#) property

[visibility](#) property

[background](#) properties

[border](#) properties

c, Multiple Col Elements

- Nếu bạn muốn tạo style cho nhiều cột hơn với các style khác nhau, sử dụng nhiều thẻ <col> hơn trong <colgroup>.

Code	Result																																			
<pre> <!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </pre>	<p>Multiple Col Elements</p> <p>Add multiple col elements in the colgroup:</p> <table border="1"> <thead> <tr> <th>MON</th><th>TUE</th><th>WED</th><th>THU</th><th>FRI</th><th>SAT</th><th>SUN</th></tr> </thead> <tbody> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr> <td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr> <tr> <td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr> <tr> <td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr> </tbody> </table>	MON	TUE	WED	THU	FRI	SAT	SUN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
MON	TUE	WED	THU	FRI	SAT	SUN																														
1	2	3	4	5	6	7																														
8	9	10	11	12	13	14																														
15	16	17	18	19	20	21																														
22	23	24	25	26	27	28																														

Thực tập Kiến trúc và Giao thức IoT

```
</head>
<body>

<h2>Multiple Col Elements</h2>
<p>Add multiple col elements
in the colgroup:</p>

<table style="width: 100%;">
  <colgroup>
    <col span="2"
style="background-color:
#D6EEEE">
    <col span="3"
style="background-color:
pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
    <th>FRI</th>
    <th>SAT</th>
    <th>SUN</th>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
    <td>4</td>
    <td>5</td>
    <td>6</td>
    <td>7</td>
  </tr>
  <tr>
    <td>8</td>
    <td>9</td>
    <td>10</td>
    <td>11</td>
    <td>12</td>
    <td>13</td>
    <td>14</td>
  </tr>
  <tr>
    <td>15</td>
    <td>16</td>
    <td>17</td>
    <td>18</td>
    <td>19</td>
    <td>20</td>
```

```

<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
</table>

</body>
</html>

```

d, Empty Colgroups

Code	Result																																			
<pre> <!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <h2>Empty Colgroups</h2> <p>Add "empty" col elements that represents the columns before the columns you want to style:</p> <table style="width: 100%;"> <colgroup> <col span="3"> <col span="2" style="background-color: pink"> </colgroup> <tr> <th>MON</th> <th>TUE</th> <th>WED</th> <th>THU</th> </pre>	<p>Empty Colgroups</p> <p>Add "empty" col elements that represents the columns before the columns you want to style:</p> <table border="1"> <thead> <tr> <th>MON</th><th>TUE</th><th>WED</th><th>THU</th><th>FRI</th><th>SAT</th><th>SUN</th></tr> </thead> <tbody> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr> <td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr> <tr> <td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr> <tr> <td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr> </tbody> </table>	MON	TUE	WED	THU	FRI	SAT	SUN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
MON	TUE	WED	THU	FRI	SAT	SUN																														
1	2	3	4	5	6	7																														
8	9	10	11	12	13	14																														
15	16	17	18	19	20	21																														
22	23	24	25	26	27	28																														

Thực tập Kiến trúc và Giao thức IoT

```
<th>FRI</th>
<th>SAT</th>
<th>SUN</th>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
</table>

</body>
</html>
```

e, Hide Columns

Code	Result																				
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <h2>Hide Columns</h2> <p>You can hide specific columns with the visibility property:</p> <table style="width: 100%;"> <colgroup> <col span="2"> <col span="3" style="visibility: collapse"> </colgroup> <tr> <th>MON</th> <th>TUE</th> <th>WED</th> <th>THU</th> <th>FRI</th> <th>SAT</th> <th>SUN</th> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td></pre>	<p>Hide Columns</p> <p>You can hide specific columns with the visibility property:</p> <table border="1" data-bbox="720 422 1176 586"> <thead> <tr> <th data-bbox="736 422 827 451">MON</th><th data-bbox="827 422 917 451">TUE</th><th data-bbox="917 422 1008 451">SAT</th><th data-bbox="1008 422 1098 451">SUN</th></tr> </thead> <tbody> <tr> <td data-bbox="736 451 827 480">1</td><td data-bbox="827 451 917 480">2</td><td data-bbox="917 451 1008 480">6</td><td data-bbox="1008 451 1098 480">7</td></tr> <tr> <td data-bbox="736 480 827 509">8</td><td data-bbox="827 480 917 509">9</td><td data-bbox="917 480 1008 509">13</td><td data-bbox="1008 480 1098 509">14</td></tr> <tr> <td data-bbox="736 509 827 539">15</td><td data-bbox="827 509 917 539">16</td><td data-bbox="917 509 1008 539">20</td><td data-bbox="1008 509 1098 539">21</td></tr> <tr> <td data-bbox="736 539 827 568">22</td><td data-bbox="827 539 917 568">23</td><td data-bbox="917 539 1008 568">27</td><td data-bbox="1008 539 1098 568">28</td></tr> </tbody> </table> <p>Note: The table columns does not collapse properly in Safari browsers.</p>	MON	TUE	SAT	SUN	1	2	6	7	8	9	13	14	15	16	20	21	22	23	27	28
MON	TUE	SAT	SUN																		
1	2	6	7																		
8	9	13	14																		
15	16	20	21																		
22	23	27	28																		

```
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
</table>

<p><b>Note:</b> The table
columns does not collapse
properly in Safari
browsers.</p>
</body>
</html>
```

11. HTML Lists

11.1. List

- List cho phép người lập trình tập hợp các phần item liên quan với nhau.

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<body>

<h2>An Unordered HTML List</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

<h2>An Ordered HTML List</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

An Unordered HTML List

- Coffee
- Tea
- Milk

An Ordered HTML List

1. Coffee
2. Tea
3. Milk

11.2. Unordered HTML List

- Danh sách không có thứ tự bắt đầu với thẻ ``. Mỗi list item bắt đầu với thẻ ``

Code	Result
<!DOCTYPE html> <html> <body> <h2>An unordered HTML list</h2> Coffee Tea Milk </body> </html>	<h2>An unordered HTML list</h2> <ul style="list-style-type: none">• Coffee• Tea• Milk

11.3. Ordered HTML List

- Danh sách có thứ tự được bắt đầu bằng thẻ ``. Mỗi phần tử bắt đầu với thẻ ``.

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<body>

<h2>An ordered HTML list</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

An ordered HTML list

1. Coffee
2. Tea
3. Milk

11.4. HTML Description Lists

- HTML cũng hỗ trợ danh sách có mô tả.
- Danh sách mô tả được sử dụng cho các thuật ngữ, giúp mô tả cụ thể từng thuật ngữ.
- Thủ `<dd>` hiển thị mô tả cho thuật ngữ, thủ `<dt>` định nghĩa tên thuật ngữ, thủ `<dt>` định nghĩa một danh sách có mô tả.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>A Description List</h2> <dl> <dt>Coffee</dt> <dd>- black hot drink</dd> <dt>Milk</dt> <dd>- white cold drink</dd> </dl> </body> </html></pre>	<p>A Description List</p> <p>Coffee - black hot drink</p> <p>Milk - white cold drink</p>

12. HTML Block & Inline

12.1. Block-level Elements

- Một block sẽ luôn bắt đầu bằng một hàng mới, và trình duyệt tự động chèn thêm một số không gian trước và sau phần tử block (gọi là margin)
- Một block luôn lấy tối đa chiều ngang nó có thể.
- Có hai thủ dùng để tạo block là `<p>` và `<div>`:
 - + Thủ `<p>` được dùng để tạo ra một đoạn văn trong tài liệu HTML

Thực tập Kiến trúc và Giao thức IoT

- + Thủ `<div>` được dùng để tạo ra các phần chia cho tài liệu HTML

Code	Result
<pre><!DOCTYPE html> <html> <body> <p style="border: 1px solid black">Hello World</p> <div style="border: 1px solid black">Hello World</div> <p>The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).</p> </body> </html></pre>	<p>Hello World</p> <p>Hello World</p> <p>The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).</p>

- Ngoài `<p>` và `<div>` còn có các block element khác trong tài liệu HTML ví dụ như:

```
<address>    <article>    <aside>    <blockquote>  <canvas>    <dd>
<div>        <dl>        <dt>        <fieldset>   <figcaption> <figure>
<footer>     <form>     <h1>-<h6>  <header>    <hr>       <li>
<main>        <nav>      <noscript>  <ol>        <p>         <pre>
<section>     <table>     <tfoot>    <ul>        <video>
```

12.2. Inline Elements

- Phần tử inline sẽ không tạo ra một hàng mới, chỉ sử dụng đủ chiều dài mà nó cần thiết.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>This is an inline span Hello World element inside a paragraph.</p> <p>The SPAN element is an inline element, and will not</pre>	<p>This is an inline span Hello World element inside a paragraph.</p> <p>The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.</p>

Thực tập Kiến trúc và Giao thức IoT

```
start on a new line and only  
takes up as much width as  
necessary.</p>  
  
</body>  
</html>
```

- Ngoài ra còn có các thẻ để tạo phần tử inline:

<a>	<abbr>	<acronym>		<bdo>	<big>
 	<button>	<cite>	<code>	<dfn>	
<i>		<input>	<kbd>	<label>	<map>
<object>	<output>	<q>	<samp>	<script>	<select>
<small>			<sub>	<sup>	<textarea>
<time>	<tt>	<var>			

12.3. The <div> Element

- Có thể thay đổi thuộc tính của cả <div> cho các phần tử bên trong nó:

Code	Result
<pre><!DOCTYPE html> <html> <body> <div style="background- color:black;color:white;padding:20px;"> <h2>London</h2> <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p> <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p> </div> </body> </html></pre>	<p>London</p> <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p> <p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>

12.4. The Element

- Thẻ được dùng để đánh dấu một phần của nội dung hoặc 1 phần của tài liệu. Thẻ không yêu cầu thuộc tính nhưng các thuộc tính như style , class và id là chung.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>The span element</h1> <p>My mother has blue eyes and my father has dark green eyes.</p> </body> </html></pre>	<h2>The span element</h2> <p>My mother has blue eyes and my father has dark green eyes.</p>

13. HTML Iframes

13.1. HTML Iframe Syntax

- Cú pháp: `<iframe src="url" title="description"></iframe>`
 - + url là đường dẫn đến trang web muốn chèn vào;
 - + description là mô tả trang web đó.

13.2. Iframe - Set Height and Width

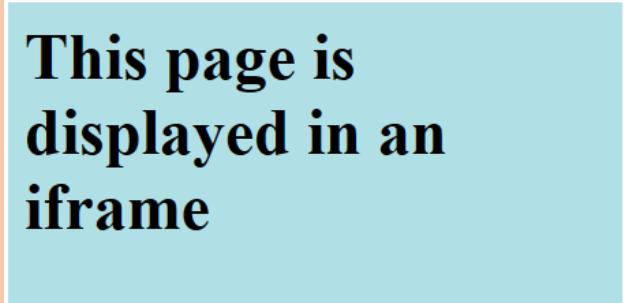
- Sử dụng các thuộc tính height và width để định dạng kích cỡ cho trang web được nhúng vào để phù hợp với bố cục trang web chính.

- Cú pháp: `<iframe src="demo_iframe.htm" height="200" width="300"
title="Iframe Example"></iframe>`

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>HTML Iframes</h2> <p>You can use the height and width attributes to specify the size of the iframe:</p> <iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe> </body> </html></pre>	<p>HTML Iframes</p> <p>You can use the height and width attributes to specify the size of the iframe:</p> <div style="background-color: #e0f2f1; padding: 10px; text-align: center;"> <p>This page is displayed in an iframe</p> </div>

13.3. Iframe - Remove the Border

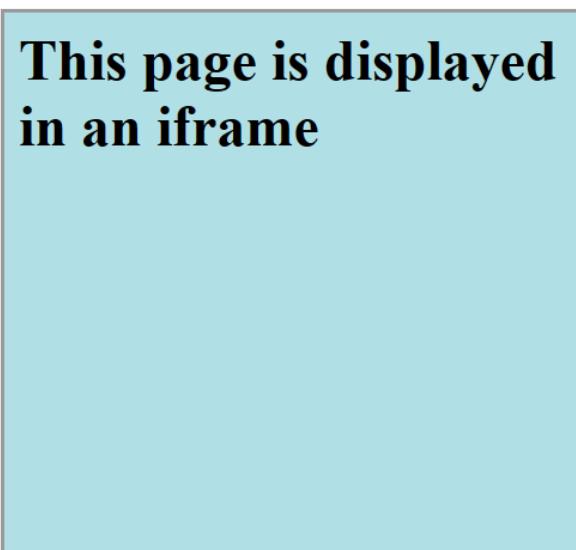
- Mặc định, một iframe có một đường viền bao quanh nó.
- Để loại bỏ đường viền có thể dùng thuộc tính style và thay đổi nội dung của border.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Remove the Iframe Border</h2> <p>To remove the default border of the iframe, use CSS:</p> <iframe src="demo_iframe.htm" style="border:none;" title="Iframe Example"></iframe> </body> </html></pre>	<p>Remove the Iframe Border</p> <p>To remove the default border of the iframe, use CSS:</p> 

- Ngoài ra, CSS còn có thể giúp thay đổi kích thước, định dạng và màu của khung viền của trang web được nhúng vào.

13.4. Iframe - Target for a Link

- Một Iframe có thể được sử dụng để làm frame đích cho một liên kết

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Iframe - Target for a Link</h2> <iframe src="demo_iframe.htm" name="iframe_a" height="300px" width="100%" title="Iframe Example"></iframe> <p>W3Schools.com</p> <p>When the target attribute of a link matches the name of an iframe, the link will open in the iframe.</p> </body> </html></pre>	<p>Iframe - Target for a Link</p>  <p><u>W3Schools.com</u></p> <p>When the target attribute of a link matches the name of an iframe, the link will open in the iframe.</p>

Thực tập Kiến trúc và Giao thức IoT

- Khi nhấn vào liên kết trong trang web, trang web được tự động chuyển đến đầu của iframe vì nó được liên kết trực tiếp đến iframe thông qua thuộc tính target với tên iframe_a.

14. HTML Javascript

- Dùng hàm Date() để lấy giá trị ngày giờ hiện tại để lưu vào đối tượng có Id “demo”

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>My First JavaScript</h1> <button type="button" onclick="document.getElementById('demo'). innerHTML = Date()"> Click me to display Date and Time.</button> <p id="demo"></p> </body> </html></pre>	<p>My First JavaScript</p> <p>Click me to display Date and Time.</p>

14.1. The HTML <script> Tag

- Đoạn mã javascript giữa hai thẻ <script> và </script> có tác dụng thay đổi phần tử có Id “demo” trong file HTML thành dòng chữ “Hello Javascript”

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Use JavaScript to Change Text</h2> <p>This example writes "Hello JavaScript!" into an HTML element with id="demo":</p> <p id="demo"></p> <script> document.getElementById("demo").innerHTML = "Hello JavaScript!"; </script> </body> </html></pre>	<p>Use JavaScript to Change Text</p> <p>This example writes "Hello JavaScript!" into an HTML element with id="demo".</p> <p>Hello JavaScript!</p>

14.2. A Taste of JavaScript

- Thuộc tính sự kiện onclick trong button “Click Me” có tác dụng gọi hàm myFunction() được viết giữa hai thẻ <script></script>.

- Hàm myFunction() có tác dụng thay đổi phần tử có Id “demo” thành nội dung là “Hello Javascript”.

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>My First JavaScript</h1> <p>JavaScript can change the content of an HTML element:</p> <button type="button" onclick="myFunction()">Click Me!</button> <p id="demo">This is a demonstration.</p> <script> function myFunction() { document.getElementById("demo"). innerHTML = "Hello JavaScript!"; } </script> </body> </html></pre>	<p>My First JavaScript</p> <p>JavaScript can change the content of an HTML element:</p> <p>Click Me!</p> <p>This is a demonstration.</p>

14.3. The HTML <noscript> Tag

- Thẻ <noscript></noscript> có tác dụng hiển thị nếu đoạn mã javascript trong cặp thẻ <script></script> không được thực thi (vì lí do trình duyệt không hỗ trợ).

Code	Result
<pre><!DOCTYPE html> <html> <body> <p id="demo"></p> <script> document.getElementById("demo"). innerHTML = "Hello JavaScript!"; </script></pre>	<p>Hello JavaScript!</p> <p>A browser without support for JavaScript will show the text written inside the noscript element.</p>

```
</script>

<noscript>Sorry, your browser does
not support JavaScript!</noscript>

<p>A browser without support for
JavaScript will show the text
written inside the noscript
element.</p>

</body>
</html>
```

- Đối với các trình duyệt không hỗ trợ dòng chữ “Sorry, your browser does not support JavaScript!” sẽ được hiện ra

15. HTML Head

- Thẻ `<head>` được đặt giữa thẻ `<html>` và thẻ `<body>`.

15.1. The HTML `<title>` Element

- Thẻ `<title>` được dùng để đặt các tiêu đề cho trang web. Nội dung của thẻ `<title>` bắt buộc phải là text.
- Thẻ `<title>` là bắt buộc cho một trang web HTML

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>A Meaningful Page Title</title> </head> <body> <p>The content of the body element is displayed in the browser window.</p> <p>The content of the title element is displayed in the browser tab, in favorites and in search-engine results.</p> </body> </html></pre>	<p>The content of the body element is displayed in the browser window.</p> <p>The content of the title element is displayed in the browser tab, in favorites and in search-engine results.</p>

15.2. The HTML `<style>` Element

- Thẻ `<style>` được dùng để định dạng cho nội dung trong trang web

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
    <style>
        body {background-color: powderblue;}
        h1 {color: red;}
        p {color: blue;}
    </style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

<p>The content of the body element is displayed in the browser window.</p>
<p>The content of the title element is displayed in the browser tab, in favorites and in search-engine results.</p>

</body>
</html>
```

This is a Heading

This is a paragraph.

The content of the body element is displayed in the browser window.

The content of the title element is displayed in the browser tab, in favorites and in search-engine results.

15.3. The HTML `<link>` Element

- Thẻ `<link>` có tác dụng xác định mối quan hệ giữa tài liệu hiện tại và một tài liệu bên ngoài

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>Page Title</title> <link rel="stylesheet" href="mystyle.css"> </head> <body> <h1>This is a Heading</h1> <p>This is a paragraph.</p> </body> </html></pre>	<h1>This is a Heading</h1> <p>This is a paragraph.</p>

15.4. The HTML <meta> Element

- Thẻ <meta> thường dùng để mô tả loại kí tự, mô tả trang web, từ khóa, tác giả của trang web và cài đặt hiển thị

- Nội dung này không được hiển thị trên trang web nhưng được sử dụng bởi trình duyệt, công cụ tìm kiếm và các dịch vụ web khác

- Dùng thẻ meta để định nghĩa loại kí tự được dùng: `<meta charset="UTF-8">`

- Dùng thẻ meta để định nghĩa keywords cho công cụ tìm kiếm:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

- Dùng thẻ meta để định nghĩa mô tả cho trang web:

```
<meta name="description" content="Free Web tutorials">
```

- Dùng thẻ meta để định nghĩa tác giả của trang web:

```
<meta name="author" content="John Doe">
```

- Dùng thẻ meta để làm mới trang web mỗi 30 giây:

```
<meta http-equiv="refresh" content="30">
```

15.5. Setting The Viewport

- Viewport giúp trang web hiển thị tốt hơn kể cả đối với trình duyệt mobile hay của desktop.

- Cú pháp:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

+ “width=device-width”: giúp trang web phù hợp với kích thước màn hình của thiết bị mà nó hiển thị.

+ “initial-scale=1.0”: giúp trang web cài mặc định mức độ zoom khi vừa truy cập là 1.0.

15.6. The HTML <script> Element

- Thẻ <script> có tác dụng giúp nhúng các đoạn mã javascript vào trang web, giúp tạo web động và tương tác tốt hơn với người dùng.

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>Page Title</title> <script> function myFunction() { document.getElementById("demo"). innerHTML = "Hello JavaScript!"; } </script> </head> <body> <h1>My Web Page</h1> <p id="demo">A Paragraph</p></pre>	<p>My Web Page</p> <p>A Paragraph</p> <p>Try it</p>

```
<button type="button"
onclick="myFunction()">Try
it</button>

</body>
</html>
```

15.7. The HTML <base> Element

- Thủ <base> có tác dụng chỉ định url cơ sở cho tất cả các url còn lại trong trang web

Code	Result
<pre><!DOCTYPE html> <html> <head> <base href="https://www.w3schools.com/" target="_blank"> </head> <body> <h1>The base element</h1> <p> - Notice that we have only specified a relative address for the image. Since we have specified a base URL in the head section, the browser will look for the image at "https://www.w3schools.com/images/stickman.gif".</p> <p>HTML base tag - Notice that the link opens in a new window, even if it has no target="_blank" attribute. This is because the target attribute of the base element is set to "_blank".</p> </body> </html></pre>	<p>The base element</p>  <p>- Notice that we have only specified a relative address for the image. Since we have specified a base URL in the head section, the browser will look for the image at "https://www.w3schools.com/images/stickman.gif".</p> <p>HTML base tag - Notice that the link opens in a new window, even if it has no target="_blank" attribute. This is because the target attribute of the base element is set to "_blank".</p>

16. HTML Layout

16.1. HTML Layout Elements

Thuộc tính	Mô tả
<header>	xác định header cho một trang web
<nav>	xác định một tập hợp các liên kết có liên quan
<section>	xác định một section trong trang web

Thực tập Kiến trúc và Giao thức IoT

<article>	xác định một nội dung độc lập trong trang web
<aside>	xác định các nội dung bên ngoài nội dung chính. Giống như thanh cuộn bên
<footer>	xác định nội dung cuối trang web như tác giả, liên hệ.

16.2. HTML Layout Techniques

- Có 4 công nghệ khác nhau để tạo ra trang web với bố cục nhiều cột. Mỗi công nghệ có ưu điểm và nhược điểm khác nhau:

a, CSS framework

- Nếu muốn tạo bố cục nhanh có thể dùng CSS framework, từ W3.CSS hoặc Bootstrap

b, CSS float property

- Ưu điểm: dễ học, chỉ cần nhớ các thuộc tính float và clear hoạt động.

- Nhược điểm: Các phần tử floating bị ràng buộc, có thể ảnh hưởng tính linh hoạt của trang web.

c, CSS flexbox

- Ưu điểm: đảm bảo các thành phần hoạt động có thể dự đoán được bố cục để phù hợp với kích thước màn hình và các thiết bị hiển thị khác nhau.

- Nhược điểm: quá phức tạp đối với các bố cục phức tạp. Gây khó khăn trong việc xử lý các thành phần không đồng nhất, không hoạt động trên các trình duyệt cũ.

d, CSS grid

- Ưu điểm: cung cấp các mạng lưới với các hàng và cột giúp thiết kế trang web dễ dàng hơn mà không cần sử dụng float và định vị

- Nhược điểm: Khó khăn trong việc làm việc với các thành phần không có kích thước cố định, có nhiều khái niệm khá phức tạp.

17. HTML Symbols

- Nhiều kí hiệu toán học, kí thuật, tiền tệ không được biểu diễn trong một bàn phím thông thường.

- Để thêm những biểu tượng như vậy vào trang web HTML, có thể sử dụng tên hoặc kí hiệu cho nó để thêm vào.

- Một số biểu thức toán học được hỗ trợ trong HTML (link full math symbols):

+ https://www.w3schools.com/charsets/ref_utf_math.asp

- Một số biểu thức Greek Letters được hỗ trợ bởi HTML (link full Greek Letters Symbols): https://www.w3schools.com/charsets/ref_utf_greek.asp

- Một số biểu thức khác được hỗ trợ bởi HTML (Currency, Arrows, Full Symbols):

+ https://www.w3schools.com/charsets/ref_utf_currency.asp

+ https://www.w3schools.com/charsets/ref_utf_arrows.asp

+ https://www.w3schools.com/charsets/ref_utf_symbols.asp

18. HTML Emoji

- UTF - 8 Characters: Nhiều ký tự UTF-8 không thể gõ được trên bàn phím nhưng chúng luôn có thể được hiển thị bằng số (được gọi là số thực thể):

Code	Result
<pre><!DOCTYPE html> <html> <meta charset="UTF-8"> <body> <h1>Sized Emojis</h1> <p style="font-size:48px"> &#128512; &#128516; &#128525; &#128151; </p> </body> </html></pre>	<p>Sized Emojis</p> 

19. HTML URL Encode

- URL là một từ khác chỉ địa chỉ web, có thể bao gồm các từ (ví dụ: w3schools.com) hoặc địa chỉ Giao thức Internet (IP) (ví dụ: 192.68.20.50).

19.1. URL - Uniform Resource Locator

- Trình duyệt web yêu cầu các trang từ máy chủ web bằng cách sử dụng URL.
 - Bộ định vị tài nguyên thống nhất (URL) được sử dụng để đánh địa chỉ một tài liệu (hoặc dữ liệu khác) trên web.
 - Địa chỉ web luôn tuân theo các quy tắc cú pháp sau:
- scheme://prefix.domain:port/path/filename
- Trong đó:
 - + scheme - xác định loại dịch vụ Internet (phổ biến nhất là http hoặc https);
 - + prefix - xác định tiền tố tên miền (mặc định cho http là www);
 - + domain - xác định tên miền Internet (như w3schools.com);
 - + port - xác định số cổng tại máy chủ (mặc định cho http là 80);
 - + path - xác định đường dẫn tại máy chủ (Nếu bỏ qua: thư mục gốc của trang web);
 - + filename - xác định tên của một tài liệu hoặc tài nguyên.

19.2. Các URL scheme phổ biến:

scheme	Viết tắt của	Đối tượng sử dụng
http	HyperText Transfer Protocol	Các trang web phổ biến. Không được mã hóa
https	Secure HyperText Transfer Protocol	Bảo mật các trang web. Đã mã hóa

ftp	File Transfer Protocol	Tải xuống hoặc tải lên tập tin
file		Một tập tin trên máy tính của bạn

19.3. Mã hóa URL

- URL chỉ có thể được gửi qua Internet bằng bộ ký tự ASCII. Nếu một URL chứa các ký tự nằm ngoài bộ ASCII thì URL đó phải được chuyển đổi.
- Mã hóa URL chuyển đổi các ký tự không phải ASCII thành định dạng có thể truyền qua Internet.
- Mã hóa URL thay thế các ký tự không phải ASCII bằng "%" theo sau là các chữ số thập lục phân.
- URL không thể chứa dấu cách. Mã hóa URL thường thay thế khoảng trắng bằng dấu cộng (+) hoặc %20
- Ví dụ: Hello Sỹ = Hello+S%E1%BB%B9

20. HTML Forms

- Một biểu mẫu HTML được sử dụng để thu thập thông tin đầu vào của người dùng. Đầu vào của người dùng thường được gửi đến máy chủ để xử lý.

20.1. Phần tử <form>

- Phần tử <form> là nơi chứa các loại phần tử đầu vào khác nhau, chẳng hạn như: trường văn bản, hộp kiểm, nút radio, nút gửi, v.v.

20.2. Phần tử <input>

- Phần tử <input> có thể được hiển thị theo nhiều cách, tùy thuộc vào thuộc tính type.

Type	Mô tả
<input type="text">	Hiển thị trường nhập văn bản một dòng. Lưu ý rằng bản thân biểu mẫu không hiển thị và độ rộng mặc định của trường nhập văn bản là 20 ký tự.
<input type="radio">	Hiển thị nút radio (để chọn một trong nhiều lựa chọn)
<input type="checkbox">	Hiển thị hộp kiểm (để chọn không hoặc nhiều lựa chọn)
<input type="submit">	Hiển thị nút gửi (để gửi biểu mẫu)
<input type="button">	Hiển thị nút có thể nhấp

20.3. Trường văn bản

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>

<h2>Text input fields</h2>

<form>
  <label for="fname">First
name:</label><br>
  <input type="text" id="fname"
name="fname" value="John"><br>
  <label for="lname">Last
name:</label><br>
  <input type="text" id="lname"
name="lname" value="Doe">
</form>

<p>Note that the form itself is
not visible.</p>

<p>Also note that the default
width of text input fields is 20
characters.</p>

</body>
</html>

```

Text input fields

First name:

Last name:

Note that the form itself is not visible.

Also note that the default width of text input fields is 20 characters.

20.4. Phần tử `<label>`

- Thẻ `<label>` xác định nhãn cho nhiều thành phần của biểu mẫu.
- Phần tử `<label>` rất hữu ích cho người dùng trình đọc màn hình vì trình đọc màn hình sẽ đọc to nhãn khi người dùng tập trung vào phần tử đầu vào.
- Phần tử `<label>` cũng giúp người dùng gấp khó khăn khi nhấp vào các vùng rất nhỏ (chẳng hạn như nút radio hoặc hộp kiểm) - bởi vì khi người dùng nhấp vào văn bản trong phần tử `<label>`, nó sẽ chuyển đổi nút radio/hộp kiểm.
- Thuộc tính `for` của thẻ `<label>` phải bằng thuộc tính `id` của phần tử `<input>` để liên kết chúng lại với nhau.
- Lưu ý: mỗi trường `input` phải có thuộc tính tên để được gửi.
- Nếu thuộc tính tên bị bỏ qua, giá trị của trường đầu vào sẽ không được gửi đi.

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<body>

<h2>The name Attribute</h2>

<form action="/action_page.php">
  <label for="fname">First
name:</label><br>
  <input type="text" id="fname"
value="John"><br><br>
  <input type="submit" value="Submit">
</form>

<p>If you click the "Submit" button,
the form-data will be sent to a page
called "/action_page.php".</p>

<p>Notice that the value of the "First
name" field will not be submitted,
because the input element does not
have a name attribute.</p>

</body>
</html>
```

The name Attribute

First name:

John

Submit

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

Notice that the value of the "First name" field will not be submitted, because the input element does not have a name attribute.

21. HTML Graphics

21.1 HTML Canvas

- Phần tử HTML <canvas> được sử dụng để vẽ đồ họa trên trang web.
- Phần tử HTML <canvas> được sử dụng để vẽ đồ họa một cách nhanh chóng thông qua JavaScript.
 - Phần tử <canvas> chỉ là nơi chứa đồ họa. Bạn phải sử dụng JavaScript để thực sự vẽ đồ họa.
 - Canvas có một số phương pháp để vẽ đường dẫn, hộp, hình tròn, văn bản và thêm hình ảnh.
 - Canvas là một vùng hình chữ nhật trên trang HTML. Theo mặc định, canvas không có đường viền và không có nội dung.
 - Tích hợp JavaScript
 - + Vẽ đường thẳng:

Code

Result

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<body>

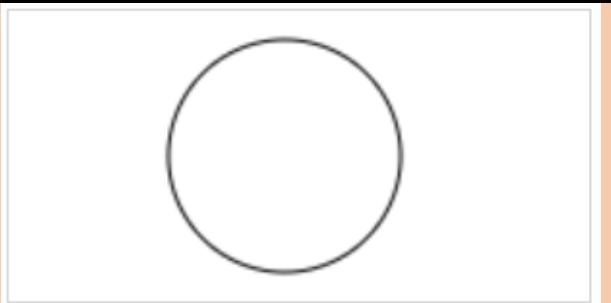
<canvas id="myCanvas" width="200"
height="100" style="border:1px solid
#d3d3d3;">
Your browser does not support the HTML
canvas tag.</canvas>

<script>
var c =
document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>

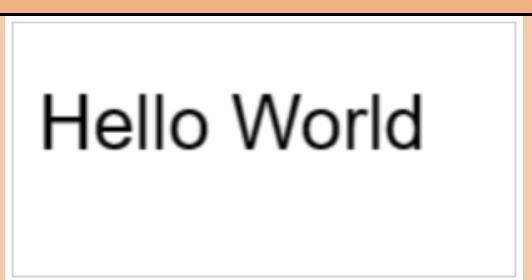
</body>
</html>
```



+ Vẽ vòng tròn:

Code	Result
<pre><script> var c = document.getElementById("myCanvas"); var ctx = c.getContext("2d"); ctx.beginPath(); ctx.arc(95, 50, 40, 0, 2 * Math.PI); ctx.stroke(); </script></pre>	

+ Vẽ văn bản:

Code	Result
<pre><script> var c = document.getElementById("myCanvas"); var ctx = c.getContext("2d"); ctx.font = "30px Arial"; ctx.fillText("Hello World", 10, 50); </script></pre>	

Thực tập Kiến trúc và Giao thức IoT

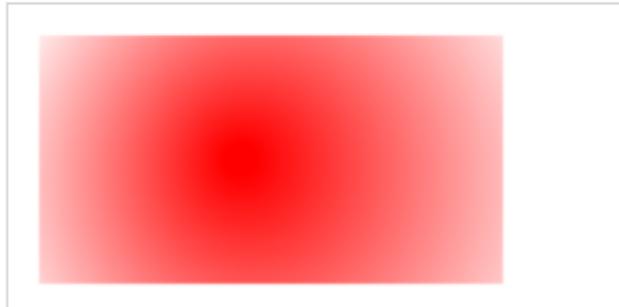
+ Viết văn bản:

Code	Result
<pre><script> var c = document.getElementById("myCanvas"); var ctx = c.getContext("2d"); ctx.font = "30px Arial"; ctx.strokeText("Hello World", 10, 50); </script></pre>	

+ Vẽ gradient tuyến tính:

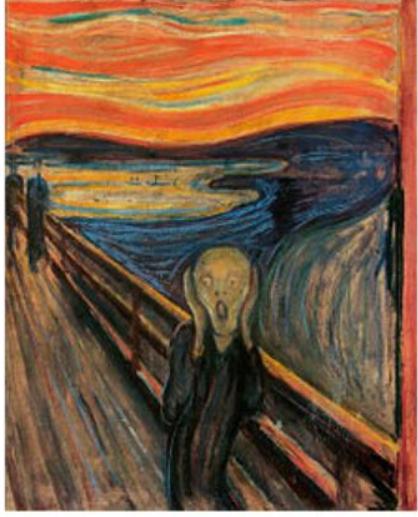
Code	Result
<pre><script> var c = document.getElementById("myCanvas"); var ctx = c.getContext("2d"); // Create gradient var grd = ctx.createLinearGradient(0, 0, 200, 0); grd.addColorStop(0, "red"); grd.addColorStop(1, "white"); // Fill with gradient ctx.fillStyle = grd; ctx.fillRect(10, 10, 150, 80); </script></pre>	

+ Vẽ gradient tròn:

Code	Result
<pre><script> var c = document.getElementById("myCanvas"); var ctx = c.getContext("2d"); // Create gradient var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100); grd.addColorStop(0, "red"); grd.addColorStop(1, "white"); // Fill with gradient ctx.fillStyle = grd; ctx.fillRect(10, 10, 150, 80); </script></pre>	

Thực tập Kiến trúc và Giao thức IoT

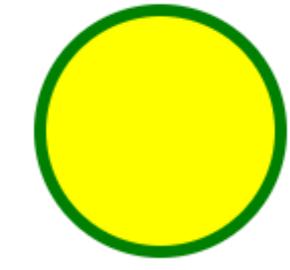
+ Vẽ ảnh:

Code	Result	
<pre><script> var c = document.getElementById("myCanvas"); var ctx = c.getContext("2d"); var img = document.getElementById("scream"); ctx.drawImage(img, 10, 10); </script></pre>	 A reproduction of Edvard Munch's famous painting "The Scream". It depicts a figure with a pale face and a wide-open mouth, screaming in anguish, set against a background of a red and orange sky over a dark, craggy landscape.	

21.2 HTML SVG

- SVG định nghĩa đồ họa dựa trên vector ở định dạng XML;
- SVG là viết tắt của Đồ họa vectơ có thể mở rộng;
- SVG được sử dụng để xác định đồ họa cho Web;
- SVG là khuyến nghị của W3C.
- SVG có một số phương pháp để vẽ đường dẫn, hộp, hình tròn, văn bản và hình ảnh đồ họa.

+ SVG Circle:

Code	Result	
<pre><!DOCTYPE html> <html> <body> <svg width="100" height="100"> <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" /> </svg> </body> </html></pre>	 A simple yellow circle with a thick green outline, centered on a white background.	

+ SVG Rectangle:

Code	Result
------	--------

Thực tập Kiến trúc và Giao thức IoT

```
<svg width="400" height="100">
    <rect width="400" height="100"
style="fill:rgb(0,0,255);stroke-
width:10;stroke:rgb(0,0,0)" />
</svg>
```



+ SVG Rounded Rectangle:

Code	Result
<pre><svg width="400" height="180"> <rect x="50" y="20" rx="20" ry="20" width="150" height="150" style="fill:red;stroke:black;stroke- width:5;opacity:0.5" /> </svg></pre>	

+ SVG Star:

Code	Result
<pre><svg width="300" height="200"> <polygon points="100,10 40,198 190,78 10,78 160,198" style="fill:lime;stroke:purple;stro ke-width:5;fill-rule:evenodd;" /> </svg></pre>	

+ SVG Logo:

Code	Result
------	--------

```
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1"
x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-
color:rgb(255,255,0);stop-opacity:1"
/>
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-
opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85"
ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-
size="45" font-family="Verdana" x="50"
y="86">SVG</text>
  Sorry, your browser does not
support inline SVG.
</svg>
```



22. HTML Video

22.1. HTML <video> element

Code	Result
<pre><!DOCTYPE html> <html> <body> <video width="320" height="240" controls> <source src="movie.mp4" type="video/mp4"> <source src="movie.ogg" type="video/ogg"> Your browser does not support the video tag. </video> </body> </html></pre>	

22.2. HTML <video> autoplay

Code	Result

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240"
autoplay>
  <source src="movie.mp4"
type="video/mp4">
  <source src="movie.ogg"
type="video/ogg">
  Your browser does not support
the video tag.
</video>

</body>
</html>
```



22.3. HTML Video - Methods, Properties, and Events

Code	Result
<pre><!DOCTYPE html> <html> <body> <div style="text-align:center"> <button onclick="playPause()">Play/Pause</button> <button onclick="makeBig()">Big</button> <button onclick="makeSmall()">Small</button> <button onclick="makeNormal()">Normal</button>

 <video id="video1" width="420"> <source src="mov_bbb.mp4" type="video/mp4"> <source src="mov_bbb.ogg" type="video/ogg"> Your browser does not support HTML video. </video> </div> <script> var myVideo = document.getElementById("video1"); function playPause() {</pre>	<div style="text-align: center;"> Play/Pause Big Small Normal </div> <p>Video courtesy of Big Buck Bunny.</p>

Thực tập Kiến trúc và Giao thức IoT

```
if (myVideo.paused)
    myVideo.play();
else
    myVideo.pause();
}

function makeBig() {
    myVideo.width = 560;
}

function makeSmall() {
    myVideo.width = 320;
}

function makeNormal() {
    myVideo.width = 420;
}
</script>

<p>Video courtesy of <a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>. </p>

</body>
</html>
```

22.4. HTML Video tags

Tags	Mô tả
<video>	Xác định video hoặc phim
<source>	Xác định nhiều tài nguyên đa phương tiện cho các thành phần đa phương tiện, chẳng hạn như <video> và <audio>
<track>	Xác định các đoạn văn bản trong trình phát đa phương tiện

23. HTML Audio

23.1. HTML <audio> autoplay

Code	Result
<!DOCTYPE html> <html> <body> <audio controls> <source src="horse.ogg" type="audio/ogg">	

```
<source src="horse.mp3"
type="audio/mpeg">
Your browser does not support the
audio element.
</audio>

</body>
</html>
```

23.2. HTML Audio - Methods, Properties, and Events

- HTML DOM xác định các phương thức, thuộc tính và sự kiện cho phần tử `<audio>`.
- Điều này cho phép bạn tải, phát và tạm dừng âm thanh cũng như đặt thời lượng và âm lượng.
- Ngoài ra còn có các sự kiện DOM có thể thông báo cho bạn khi âm thanh bắt đầu phát, bị tạm dừng, v.v.

23.3. HTML Audio tags

Tags	Mô tả
<code><audio></code>	Xác định nội dung âm thanh
<code><source></code>	Xác định nhiều tài nguyên đa phương tiện cho các thành phần đa phương tiện, chẳng hạn như <code><video></code> và <code><audio></code>

24. HTML Plug-ins

24.1. Plug-ins

- Plug-ins là các chương trình máy tính mở rộng chức năng tiêu chuẩn của trình duyệt.
- Plug-in được thiết kế để sử dụng cho nhiều mục đích khác nhau:
 - + Để chạy các ứng dụng Java;
 - + Để chạy các điều khiển Microsoft ActiveX;
 - + Để hiển thị phim Flash;
 - + Để hiển thị bản đồ;
 - + Để quét virus;
 - + Để xác minh id ngân hàng.
- Lưu ý: Hầu hết các trình duyệt không còn hỗ trợ Java Applet và Plug-in , Điều khiển ActiveX.
 - Hỗ trợ cho Shockwave Flash cũng đã bị tắt trong các trình duyệt hiện đại.

24.2 Phản tử `<object>`

- Phản tử `<object>` xác định một đối tượng được nhúng trong tài liệu HTML.
- Nó được thiết kế để nhúng các plug-in (như ứng dụng Java, trình đọc PDF và Trình phát Flash) vào các trang web nhưng cũng có thể được sử dụng để đưa HTML vào HTML.

Code	Result		
<!DOCTYPE html>	Alfreds Futterkiste	Berlin	Germany
<html>	Berglunds snabbköp	Luleå	Sweden
<body>	Centro comercial Moctezuma	México D.F.	Mexico
<object width="100%" height="500px" data="snippet.html"></object>	Ernst Handel	Graz	Austria
</body>	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain
</html>	Galería del gastrónomo	Barcelona	Spain
	Island Trading	Cowes	UK
	Königlich Essen	Brandenburg	Germany
	Laughing Bacchus Wine Cellars	Vancouver	Canada
	Magazzini Alimentari Riuniti	Bergamo	Italy
	North/South	London	UK
	Paris spécialités	Paris	France
	Rattlesnake Canyon Grocery	Albuquerque	USA
	Simons bistro	København	Denmark
	The Big Cheese	Portland	USA
	Vaffeljernet	Århus	Denmark
	Wolski Zajazd	Warszawa	Poland

24.3 Phản tử <embed>

- Phản tử <embed> cũng xác định một đối tượng được nhúng trong tài liệu HTML.
- Các trình duyệt web đã hỗ trợ phản tử <embed> từ lâu. Tuy nhiên, nó chưa phải là một phản của đặc tả HTML trước HTML5.
- Lưu ý: phản tử <embed> không có thẻ đóng. Nó không thể chứa văn bản thay thế. Phản tử <embed> cũng có thể được sử dụng để đưa HTML vào HTML.

Thực tập Kiến trúc và Giao thức IoT

Code	Result		
<!DOCTYPE html>	Alfreds Futterkiste	Berlin	Germany
<html>	Berglunds snabbköp	Luleå	Sweden
<body>	Centro comercial Moctezuma	México D.F.	Mexico
<embed width="100%" height="500px" src="snippet.html">	Ernst Handel	Graz	Austria
</body>	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain
</html>	Galería del gastrónomo	Barcelona	Spain
	Island Trading	Cowes	UK
	Königlich Essen	Brandenburg	Germany
	Laughing Bacchus Wine Cellars	Vancouver	Canada
	Magazzini Alimentari Riuniti	Bergamo	Italy
	North/South	London	UK
	Paris spécialités	Paris	France
	Rattlesnake Canyon Grocery	Albuquerque	USA
	Simons bistro	København	Denmark
	The Big Cheese	Portland	USA
	Vaffeljernet	Århus	Denmark
	Wolski Zajazd	Warszawa	Poland

25. HTML YouTube

Cách dễ nhất để phát video bằng HTML là sử dụng YouTube.

Code	Result
<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/tgbNymZ7vqY" > </iframe> </body> </html></pre>	

25.1 Youtube Autoplay + Mute

- Để video bắt đầu phát tự động khi người dùng truy cập trang thêm autoplay=1 vào URL YouTube.

- Lưu ý: Trình duyệt Chrome không cho phép tự động phát trong hầu hết các trường hợp. Tuy nhiên, tính năng tự động phát ở chế độ tắt tiếng luôn được cho phép.

Thêm mute=1 sau autoplay=1 để video của bạn bắt đầu phát tự động (nhưng bị tắt tiếng).

Code	Result
<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/ tgbNymZ7vqY?autoplay=1&mute=1"> </iframe> </body> </html></pre>	

25.2 Youtube Playlists

- Danh sách video sẽ phát được phân tách bằng dấu phẩy (ngoài URL gốc).

25.3 YouTube Loop

Code	Result
<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/ tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1"> </iframe> </body> </html></pre>	

25.4 YouTube Controls

- Thêm control=0: không hiển thị các điều khiển trong trình phát video, control =1 (mặc định): hiển thị điều khiển trình phát.

Code	Result
<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/ tgbNymZ7vqY?controls=0"> </iframe> </body> </html></pre>	

26. HTML Geolocation API

- HTML Geolocation API được sử dụng để xác định vị trí của người dùng.
- Sử dụng HTML Geolocation:

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>HTML Geolocation</h1> <p>Click the button to get your coordinates.</p> <button onclick="getLocation()">Try It</button> <p id="demo"></p> <script> const x = document.getElementById("demo"); function getLocation() { if (navigator.geolocation) { navigator.geolocation.getCurrentPosition(showPosition); } else { x.innerHTML = "Geolocation is not supported by this browser."; } } function showPosition(position) { x.innerHTML = "Latitude: " + position.coords.latitude + "
Longitude: " + position.coords.longitude; } </script> </body> </html></pre>	<p>HTML Geolocation</p> <p>Click the button to get your coordinates.</p> <p>Try It</p> <p>Latitude: 10.835149 Longitude: 106.776107</p>

- Phương thức getCurrentPosition() - Trả về dữ liệu

Property	Trả về
coords.latitude	Vĩ độ dưới dạng số thập phân (luôn được trả về)
coords.longitude	Kinh độ dưới dạng số thập phân (luôn được trả về)
coords.accuracy	Độ chính xác của vị trí (luôn được trả về)
coords.altitude	Độ cao tính bằng mét so với mực nước biển trung bình (trả về nếu có)

Thực tập Kiến trúc và Giao thức IoT

coords.altitudeAccuracy	Độ chính xác về độ cao của vị trí (được trả về nếu có)
coords.heading	Tiêu đề theo độ theo chiều kim đồng hồ từ hướng Bắc (trả về nếu có)
coords.speed	Tốc độ tính bằng mét trên giây (được trả về nếu có)
timestamp	Ngày/giờ của phản hồi (được trả về nếu có)

- Các phương thức khác:

+ watchPosition() - Trả về vị trí hiện tại của người dùng và tiếp tục trả về vị trí đã cập nhật khi người dùng di chuyển (như GPS trong ô tô).

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>HTML Geolocation</h1> <p>Click the button to get your coordinates.</p> <button onclick="getLocation()">Try It</button> <p id="demo"></p> <script> const x = document.getElementById("demo"); function getLocation() { if (navigator.geolocation) { navigator.geolocation.watchPosition (showPosition); } else { x.innerHTML = "Geolocation is not supported by this browser."; } } function showPosition(position) { x.innerHTML="Latitude: " + position.coords.latitude + "
Longitude: " + position.coords.longitude; } </script> </body> </html></pre>	<p>HTML Geolocation</p> <p>Click the button to get your coordinates.</p> <p>Try It</p> <p>Latitude: 10.83515 Longitude: 106.776124</p>

+ clearWatch() - Dừng phương thức watchPosition().

II. CSS

1. CSS Colors

1.1. Colors

a, CSS Color Names

Code	Result
<!DOCTYPE html> <html> <body>	Tomato
<h1 style="background-color: Tomato;">Tomato</h1>	Orange
<h1 style="background-color: Orange;">Orange</h1>	DodgerBlue
<h1 style="background-color: DodgerBlue;">DodgerBlue</h1>	MediumSeaGreen
<h1 style="background-color: MediumSeaGreen;">MediumSeaGreen</h1>	Gray
>	
<h1 style="background-color: Gray;">Gray</h1>	SlateBlue
<h1 style="background-color: SlateBlue;">SlateBlue</h1>	Violet
<h1 style="background-color: Violet;">Violet</h1>	LightGray
<h1 style="background-color: LightGray;">LightGray</h1>	
</body>	
</html>	

b, CSS Background Color

Code	Result
<!DOCTYPE html> <html> <body>	
<h1 style="background-color: DodgerBlue;">Hello World</h1>	Hello World

Below the result of the first row:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

```
Ut wisi enim ad minim veniam, quis  
nostrud exerci tation ullamcorper  
suscipit lobortis nisl ut aliquip  
ex ea commodo consequat.  
</p>  
  
</body>  
</html>
```

c, CSS Text Color

Code	Result
<pre><!DOCTYPE html> <html> <body> <h3 style="color:Tomato;">Hello World</h3> <p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p> <p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p> </body> </html></pre>	<p>Hello World</p> <p>Lore ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p> <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>

d, CSS Border Color

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1 style="border: 2px solid Tomato;">Hello World</h1> <h1 style="border: 2px solid DodgerBlue;">Hello World</h1> <h1 style="border: 2px solid Violet;">Hello World</h1> </body> </html></pre>	<p>Hello World</p> <p>Hello World</p> <p>Hello World</p>

e, CSS Color Values

Code	Result
<!DOCTYPE html> <html> <body> <p>Same as color name "Tomato":</p> <h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1> <h1 style="background-color:#ff6347;">#ff6347</h1> <h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1> <p>Same as color name "Tomato", but 50% transparent:</p> <h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h1> <h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1> <p>In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.</p> </body> </html>	Same as color name "Tomato": rgb(255, 99, 71) #ff6347 hsl(9, 100%, 64%) Same as color name "Tomato", but 50% transparent: rgba(255, 99, 71, 0.5) hsla(9, 100%, 64%, 0.5) In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.

1.2. RGB

a, RGB Value

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<body>

<h1>Specify colors using RGB
values</h1>

<h2 style="background-color:rgb(255,
0, 0);">rgb(255, 0, 0)</h2>
<h2 style="background-color:rgb(0, 0,
255);">rgb(0, 0, 255)</h2>
<h2 style="background-color:rgb(60,
179, 113);">rgb(60, 179, 113)</h2>
<h2 style="background-color:rgb(238,
130, 238);">rgb(238, 130, 238)</h2>
<h2 style="background-color:rgb(255,
165, 0);">rgb(255, 165, 0)</h2>
<h2 style="background-color:rgb(106,
90, 205);">rgb(106, 90, 205)</h2>

</body>
</html>
```

Specify colors using RGB values

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)

rgb(255, 165, 0)

rgb(106, 90, 205)

b, RGBA Value

Code	Result
<!DOCTYPE html> <html> <body> <h1>Make transparent colors with RGBA</h1> <h2 style="background-color:rgba(255, 99, 71, 0);">rgba(255, 99, 71, 0)</h2> <h2 style="background-color:rgba(255, 99, 71, 0.2);">rgba(255, 99, 71, 0.2)</h2> <h2 style="background-color:rgba(255, 99, 71, 0.4);">rgba(255, 99, 71, 0.4)</h2> <h2 style="background-color:rgba(255, 99, 71, 0.6);">rgba(255, 99, 71, 0.6)</h2> <h2 style="background-color:rgba(255, 99, 71, 0.8);">rgba(255, 99, 71, 0.8)</h2> <h2 style="background-color:rgba(255, 99, 71, 1);">rgba(255, 99, 71, 1)</h2> </body> </html>	Make transparent colors with RGBA
	rgba(255, 99, 71, 0)
	rgba(255, 99, 71, 0.2)
	rgba(255, 99, 71, 0.4)
	rgba(255, 99, 71, 0.6)
	rgba(255, 99, 71, 0.8)
	rgba(255, 99, 71, 1)

1.3. HEX

a, HEX Value

Code	Result
<!DOCTYPE html> <html> <body> <h1>Specify colors using HEX values</h1> <h2 style="background-color:#ff0000;">#ff0000</h2> <h2 style="background-color:#0000ff;">#0000ff</h2> <h2 style="background-color:#3cb371;">#3cb371</h2> <h2 style="background-color:#ee82ee;">#ee82ee</h2> <h2 style="background-color:#ffa500;">#ffa500</h2> <h2 style="background-color:#6a5acd;">#6a5acd</h2> </body> </html>	Specify colors using HEX values #ff0000 #0000ff #3cb371 #ee82ee #ffa500 #6a5acd

b, 3 Digit HEX Value

Code	Result
<!DOCTYPE html> <html> <head> <style> body { background-color: #fc9; /* same as #ffcc99 */ } h1 { color: #f0f; /* same as #ff00ff */ } p { color: #b58; /* same as #bb5588 */ } </style> </head> <body> This is a paragraph.	CSS 3-digit Hex Code This is a paragraph.

```
<h1>CSS 3-digit Hex Code</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

1.4. HSL

a, HSL value

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>Specify colors using HSL values</h1> <h2 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h2> <h2 style="background-color:hsl(240, 100%, 50%);">hsl(240, 100%, 50%)</h2> <h2 style="background-color:hsl(147, 50%, 47%);">hsl(147, 50%, 47%)</h2> <h2 style="background-color:hsl(300, 76%, 72%);">hsl(300, 76%, 72%)</h2> <h2 style="background-color:hsl(39, 100%, 50%);">hsl(39, 100%, 50%)</h2> <h2 style="background-color:hsl(248, 53%, 58%);">hsl(248, 53%, 58%)</h2> </body> </html></pre>	Specify colors using HSL values hsl(0, 100%, 50%) hsl(240, 100%, 50%) hsl(147, 50%, 47%) hsl(300, 76%, 72%) hsl(39, 100%, 50%) hsl(248, 53%, 58%)

b, HSLA value

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>Make transparent colors with HSLA</h1> <h2 style="background-color:hsla(9, 100%, 64%, 0);">hsla(9, 100%, 64%, 0)</h2> <h2 style="background-color:hsla(9, 100%, 64%, 0.2);">hsla(9, 100%, 64%, 0.2)</h2> <h2 style="background-color:hsla(9, 100%, 64%, 0.4);">hsla(9, 100%, 64%, 0.4)</h2> <h2 style="background-color:hsla(9, 100%, 64%, 0.6);">hsla(9, 100%, 64%, 0.6)</h2> <h2 style="background-color:hsla(9, 100%, 64%, 0.8);">hsla(9, 100%, 64%, 0.8)</h2> <h2 style="background-color:hsla(9, 100%, 64%, 1);">hsla(9, 100%, 64%, 1)</h2></pre>	Make transparent colors with HSLA hsla(9, 100%, 64%, 0) hsla(9, 100%, 64%, 0.2) hsla(9, 100%, 64%, 0.4) hsla(9, 100%, 64%, 0.6) hsla(9, 100%, 64%, 0.8) hsla(9, 100%, 64%, 1)

```
<h2 style="background-color:hsla(9, 100%, 64%, 0.4);>hsla(9, 100%, 64%, 0.4)</h2>
<h2 style="background-color:hsla(9, 100%, 64%, 0.6);>hsla(9, 100%, 64%, 0.6)</h2>
<h2 style="background-color:hsla(9, 100%, 64%, 0.8);>hsla(9, 100%, 64%, 0.8)</h2>
<h2 style="background-color:hsla(9, 100%, 64%, 1);>hsla(9, 100%, 64%, 1)</h2>

</body>
</html>
```

2. CSS Backgrounds

2.1. *Backgrounds color*

a, *CSS Backgrounds color*

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-color: lightblue; } </style> </head> <body> <h1>Hello World!</h1> <p>This page has a light blue background color!</p> </body> </html></pre>	<p>Hello World!</p> <p>This page has a light blue background color!</p>

b, *Opacity / Transparency*

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: green;
}

div.first {
  opacity: 0.1;
}

div.second {
  opacity: 0.3;
}

div.third {
  opacity: 0.6;
}
</style>
</head>
<body>

<h1>Transparent Boxes</h1>

<p>When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:</p>

<div class="first">
  <h1>opacity 0.1</h1>
</div>

<div class="second">
  <h1>opacity 0.3</h1>
</div>

<div class="third">
  <h1>opacity 0.6</h1>
</div>

<div>
  <h1>opacity 1 (default)</h1>
</div>

</body>
</html>
```

Transparent Boxes

When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:

opacity 0.1

opacity 0.3

opacity 0.6

opacity 1 (default)

c, Transparency using *RGBA*

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { background: rgb(0, 128, 0); } div.first { background: rgba(0, 128, 0, 0.1); } div.second { background: rgba(0, 128, 0, 0.3); } div.third { background: rgba(0, 128, 0, 0.6); } </style> </head> <body> <h1>Transparent Boxes 2</h1> <p>Result with opacity:</p> <div style="opacity:0.1;"> <h1>10% opacity</h1> </div> <div style="opacity:0.3;"> <h1>30% opacity</h1> </div> <div style="opacity:0.6;"> <h1>60% opacity</h1> </div> <div> <h1>opacity 1</h1> </div> <p>Result with rgba():</p></pre>	<p>Result with <code>rgba()</code>:</p> <p>10% opacity</p> <p>30% opacity</p> <p>60% opacity</p> <p>default</p> <p>Notice how the text gets transparent as well as the background color when using the <code>opacity</code> property.</p>

```
<div class="first">
  <h1>10% opacity</h1>
</div>

<div class="second">
  <h1>30% opacity</h1>
</div>

<div class="third">
  <h1>60% opacity</h1>
</div>

<div>
  <h1>default</h1>
</div>

<p>Notice how the text gets
transparent as well as the
background color when using the
opacity property.</p>

</body>
</html>
```

2.2. Background Image

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url("paper.gif"); } </style> </head> <body> <h1>Hello World!</h1> <p>This page has an image as the background!</p> </body> </html></pre>	<p>Hello World!</p> <p>This page has an image as the background!</p>

2.3. Background Repeat

a, CSS background-repeat

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url("gradient_bg.png"); background-repeat: repeat-x; } </style> </head> <body> <h1>Hello World!</h1> <p>Here, a background image is repeated only horizontally!</p> </body> </html></pre>	<p>Hello World!</p> <p>Here, a background image is repeated only horizontally!</p>

b, CSS background-repeat: no-repeat

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url("img_tree.png"); background-repeat: no-repeat; } </style> </head> <body> <h1>Hello World!</h1> <p>W3Schools background image example.</p> <p>The background image only shows once, but it is disturbing the reader!</p> </body> </html></pre>	<p>Hello World!</p> <p>W3Schools background image example.</p> <p>The background image only shows once, but it is disturbing the reader!</p> 

c, CSS *background-position*

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-image: url("img_tree.png"); background-repeat: no-repeat; background-position: right top; margin-right: 200px; } </style> </head> <body> <h1>Hello World!</h1> <p>Here, the background image is only shown once. In addition it is positioned away from the text.</p> <p>In this example we have also added a margin on the right side, so that the background image will not disturb the text.</p> </body> </html></pre>	<p>Hello World!</p> <p>Here, the background image is only shown once. In addition it is positioned away from the text.</p> <p>In this example we have also added a margin on the right side, so that the background image will not disturb the text.</p> 

2.4. Background-attachment

Code	Result
<pre data-bbox="165 278 838 2021"><!DOCTYPE html> <html> <head> <style> body { background-image: url("img_tree.png"); background-repeat: no-repeat; background-position: right top; margin-right: 200px; background-attachment: fixed; } </style> </head> <body> <h1>The background-attachment Property</h1> <p>The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).</p> <p>Tip: If you do not see any scrollbars, try to resize the browser window.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p> <p>The background-image is fixed. Try to scroll down the page.</p></pre>	<p data-bbox="886 309 1156 518">The background- attachment Property</p> <p data-bbox="886 548 1144 736">The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).</p> <p data-bbox="886 754 1144 837">Tip: If you do not see any scrollbars, try to resize the browser window.</p> <p data-bbox="886 860 1140 938">The background-image is fixed. Try to scroll down the page.</p> <p data-bbox="886 961 1140 1044">The background-image is fixed. Try to scroll down the page.</p> <p data-bbox="886 1062 1140 1089">The background-image is</p> 

Thực tập Kiến trúc và Giao thức IoT

```
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
<p>The background-image is fixed. Try  
to scroll down the page.</p>  
</body>  
</html>
```

2.5. Background Shorthand

Code	Result
<pre data-bbox="165 278 759 1516"><!DOCTYPE html> <html> <head> <style> body { background: #ffffff url("img_tree.png") no-repeat right top; margin-right: 200px; } </style> </head> <body> <h1>The background Property</h1> <p>The background property is a shorthand property for specifying all the background properties in one declaration.</p> <p>Here, the background image is only shown once, and it is also positioned in the top-right corner.</p> <p>We have also added a right margin, so that the text will not write over the background image.</p> </body> </html></pre>	<p data-bbox="800 323 1086 503">The background Property</p> <p data-bbox="800 534 1117 691">The background property is a shorthand property for specifying all the background properties in one declaration.</p> <p data-bbox="800 720 1117 844">Here, the background image is only shown once, and it is also positioned in the top-right corner.</p> <p data-bbox="800 873 1117 997">We have also added a right margin, so that the text will not write over the background image.</p> 

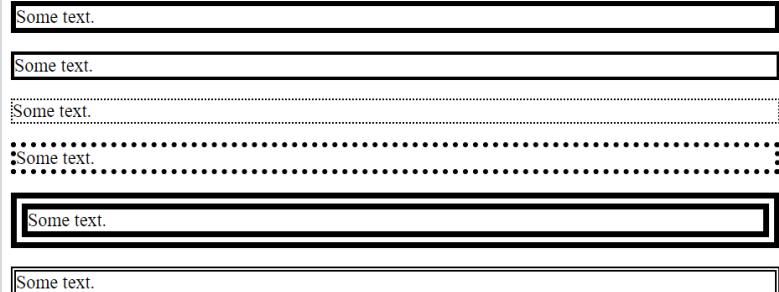
3. CSS Border

3.1. Borders

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.dotted {border-style: dotted;} p.dashed {border-style: dashed;} p.solid {border-style: solid;} p.double {border-style: double;} p.groove {border-style: groove;} p.ridge {border-style: ridge;} p.inset {border-style: inset;} p.outset {border-style: outset;} p.none {border-style: none;} p.hidden {border-style: hidden;} p.mix {border-style: dotted dashed solid double;} </style> </head> <body> <h2>The border-style Property</h2> <p>This property specifies what kind of border to display:</p> <p class="dotted">A dotted border.</p> <p class="dashed">A dashed border.</p> <p class="solid">A solid border.</p> <p class="double">A double border.</p> <p class="groove">A groove border.</p> <p class="ridge">A ridge border.</p> <p class="inset">An inset border.</p> <p class="outset">An outset border.</p> <p class="none">No border.</p> <p class="hidden">A hidden border.</p> <p class="mix">A mixed border.</p> </body> </html></pre>	<p>The border-style Property</p> <p>This property specifies what kind of border to display:</p> <ul style="list-style-type: none"> A dotted border. A dashed border. A solid border. A double border. A groove border. A ridge border. An inset border. An outset border. No border. A hidden border. A mixed border.

3.2. Border Width

a, CSS Border Width

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-width: 5px; } p.two { border-style: solid; border-width: medium; } p.three { border-style: dotted; border-width: 2px; } p.four { border-style: dotted; border-width: thick; } p.five { border-style: double; border-width: 15px; } p.six { border-style: double; border-width: thick; } </style> </head> <body> <h2>The border-width Property</h2> <p>This property specifies the width of the four borders:</p> <p class="one">Some text.</p></pre>	<p>The border-width Property</p> <p>This property specifies the width of the four borders:</p>  <p>Note: The "border-width" property does not work if it is used alone. Always specify the "border-style" property to set the borders first.</p>

Thực tập Kiến trúc và Giao thức IoT

```
<p class="two">Some  
text.</p>  
<p class="three">Some  
text.</p>  
<p class="four">Some  
text.</p>  
<p class="five">Some  
text.</p>  
<p class="six">Some  
text.</p>  
  
</body>  
</html>
```

b, Specific Side Widths

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */ } p.two { border-style: solid; border-width: 20px 5px; /* 20px top and bottom, 5px on the sides */ } p.three { border-style: solid; border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */ } </style> </head> <body> <h2>The border-width Property</h2> <p class="one">Some text.</p> <p class="two">Some text.</p> <p class="three">Some text.</p> </body> </html></pre>	<p>The border-width Property</p> 

3.3. Border Color

a, CSS Border Color

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-color: red; } p.two { border-style: solid; border-color: green; } p.three { border-style: dotted; border-color: blue; } </style> </head> <body> <h2>The border-color Property</h2> <p>This property specifies the color of the four borders:</p> <p class="one">A solid red border</p> <p class="two">A solid green border</p> <p class="three">A dotted blue border</p> <p>Note: The "border- color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p> </body> </html></pre>	<p>The border-color Property</p> <p>This property specifies the color of the four borders:</p> <p>A solid red border</p> <p>A solid green border</p> <p>A dotted blue border</p> <p>Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>

b, Specific Side Colors

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-color: red green blue yellow; /* red top, green right, blue bottom and yellow left */ } </style> </head> <body> <h2>The border-color Property</h2> <p class="one">A solid multicolor border</p> </body> </html></pre>	<p>The border-color Property</p> <p>A solid multicolor border</p>

c, HEX Values

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-color: #ff0000; /* red */ } p.two { border-style: solid; border-color: #0000ff; /* blue */ } p.three { border-style: solid; border-color: #bbbbbb; /* grey */ } </style> </head></pre>	<p>The border-color Property</p> <p>The color of the border can also be specified using a hexadecimal value (HEX):</p> <p>A solid red border</p> <p>A solid blue border</p> <p>A solid grey border</p>

```
<body>

<h2>The border-color Property</h2>
<p>The color of the border can
also be specified using a
hexadecimal value (HEX):</p>

<p class="one">A solid red
border</p>
<p class="two">A solid blue
border</p>
<p class="three">A solid grey
border</p>

</body>
</html>
```

d, RGB Values

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-color: rgb(255, 0, 0); /* red */ } p.two { border-style: solid; border-color: rgb(0, 0, 255); /* blue */ } p.three { border-style: solid; border-color: rgb(187, 187, 187); /* grey */ } </style> </head> <body> <h2>The border-color Property</h2> <p>The color of the border can also be specified using RGB values:</p></pre>	<p>The border-color Property</p> <p>The color of the border can also be specified using RGB values:</p> <p>A solid red border</p> <p>A solid blue border</p> <p>A solid grey border</p>

```
<p class="one">A solid red border</p>
<p class="two">A solid blue border</p>
<p class="three">A solid grey border</p>

</body>
</html>
```

e, HSL Values

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.one { border-style: solid; border-color: hsl(0, 100%, 50%); /* red */ } p.two { border-style: solid; border-color: hsl(240, 100%, 50%); /* blue */ } p.three { border-style: solid; border-color: hsl(0, 0%, 73%); /* grey */ } </style> </head> <body> <h2>The border-color Property</h2> <p>The color of the border can also be specified using HSL values:</p> <p class="one">A solid red border</p> <p class="two">A solid blue border</p> <p class="three">A solid grey border</p></pre>	<p>The border-color Property</p> <p>The color of the border can also be specified using HSL values:</p> <p>A solid red border</p> <p>A solid blue border</p> <p>A solid grey border</p>

```
<p class="three">A solid grey  
border</p>  
  
</body>  
</html>
```

3.4. Boder Sides

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { border-top-style: dotted; border-right-style: solid; border-bottom-style: dotted; border-left-style: solid; } </style> </head> <body> <h2>Individual Border Sides</h2> <p>2 different border styles.</p> </body> </html></pre>	<p>Individual Border Sides</p> <p>2 different border styles.</p>

3.5. Border Shorthand

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { border: 5px solid red; } </style> </head> <body> <h2>The border Property</h2> <p>This property is a shorthand property for border-width, border-style, and border- color.</p> </body> </html></pre>	<p>The border Property</p> <p>This property is a shorthand property for border-width, border-style, and border-color.</p>

3.6. Rounded Borders

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> p.normal { border: 2px solid red; padding: 5px; } p.round1 { border: 2px solid red; border-radius: 5px; padding: 5px; } p.round2 { border: 2px solid red; border-radius: 8px; padding: 5px; } p.round3 { border: 2px solid red; border-radius: 12px; padding: 5px; } </style> </head> <body> <h2>The border-radius Property</h2> <p>This property is used to add rounded borders to an element:</p> <p class="normal">Normal border</p> <p class="round1">Round border</p> <p class="round2">Rounder border</p> <p class="round3">Roundest border</p> </body> </html> </pre>	<p>The border-radius Property</p> <p>This property is used to add rounded borders to an element:</p>  <p>Normal border</p> <p>Round border</p> <p>Rounder border</p> <p>Roundest border</p>

4. CSS Margin

4.1. Margin

a, CSS Margins

- Khoảng cách (margins) được sử dụng để tạo ra khoảng trống xung quanh các phần tử, bên ngoài các đường viền đã định.

- Với CSS, bạn hoàn toàn có thể kiểm soát các khoảng cách (margins). Có các thuộc tính để đặt khoảng cách cho từng cạnh của một phần tử (trên, bên phải, dưới và bên trái). Điều này cho phép bạn điều chỉnh khoảng cách xung quanh phần tử theo ý muốn.

b, Margin - Individual Sides

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { border: 1px solid black; margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px; background-color: lightblue; } </style> </head> <body> <h2>Using individual margin properties</h2> <div>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.</div> </body> </html></pre>	<p>Using individual margin properties</p> <p>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.</p>

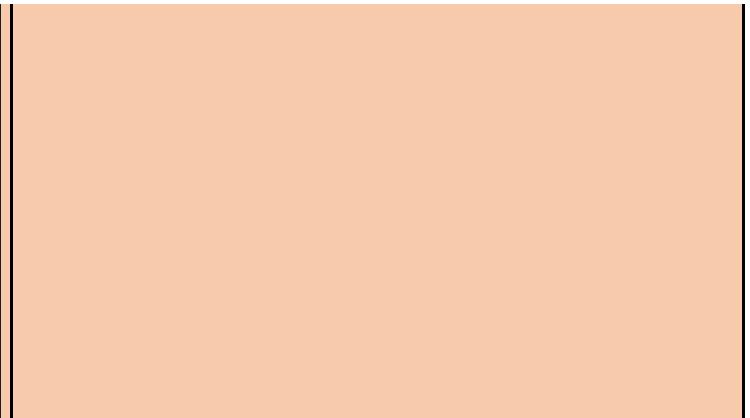
c, Margin - Shorthand Property

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { border: 1px solid black; margin: 25px 50px 75px 100px; background-color: lightblue; } </style> </head> <body> <h2>The margin shorthand property - 4 values</h2> <div>This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.</div> <hr> </body> </html></pre>	<p>The margin shorthand property - 4 values</p> <p>This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.</p>

d, The auto Value

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { width: 300px; margin: auto; border: 1px solid red; } </style> </head> <body> <h2>Use of margin: auto</h2> <p>You can set the margin property to auto to</pre>	<p>Use of margin: auto</p> <p>You can set the margin property to auto to horizontally center the element within its container.</p> <p>This div will be horizontally centered because it has margin: auto;</p>

```
horizontally center the element  
within its container.</p>  
  
<div>  
This div will be horizontally  
centered because it has margin:  
auto;  
</div>  
  
</body>  
</html>
```



e, The inherit Value

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { border: 1px solid red; margin-left: 100px; } p.ex1 { margin-left: inherit; } </style> </head> <body> <h2>Use of the inherit value</h2> <p>Let the left margin be inherited from the parent element:</p> <div> <p class="ex1">This paragraph has an inherited left margin (from the div element).</p> </div> </body> </html></pre>	<p>Use of the inherit value</p> <p>Let the left margin be inherited from the parent element:</p> <div style="border: 1px solid red; padding: 5px; margin-left: 100px;"><p>This paragraph has an inherited left margin (from the div element).</p></div>

4.2. Margin Collapse

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { margin: 0 0 50px 0; } h2 { margin: 20px 0 0 0; } </style> </head> <body> <p>Example.</p> <h1>Heading 1</h1> <h2>Heading 2</h2> </body> </html></pre>	<p>Example.</p> <p>Heading 1</p> <p>Heading 2</p>

5. CSS Padding

- Padding được sử dụng để tạo ra khoảng trống xung quanh nội dung của một phần tử, bên trong bất kỳ đường viền được định nghĩa nào.

a, Padding - Individual Sides

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { border: 1px solid black; background-color: lightblue; padding-top: 50px; padding-right: 30px; padding-bottom: 50px; padding-left: 80px; } </style> </head> <body> <h2>Using individual padding properties</h2> </pre>	<p>Using individual padding properties</p> <div style="background-color: #ADD8E6; padding: 50px 30px 50px 80px; border: 1px solid black;"> <p>This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.</p> </div>

Thực tập Kiến trúc và Giao thức IoT

```
<div>This div element has a top  
padding of 50px, a right padding of  
30px, a bottom padding of 50px, and  
a left padding of 80px.</div>  
</body>  
</html>
```

b, Padding - Shorthand Property

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { border: 1px solid black; padding: 25px 50px 75px 100px; background-color: lightblue; } </style> </head> <body> <h2>The padding shorthand property - 4 values</h2> <div>This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.</div> </body> </html></pre>	<p>The padding shorthand property - 4 values</p> <p>This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.</p>

c, Padding and Element Width

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div.ex1 { width: 300px; background-color: yellow; } div.ex2 { width: 300px; padding: 25px; background-color: lightblue; } </style></pre>	<p>Padding and element width</p> <p>This div is 300px wide.</p> <p>The width of this div is 350px, even though it is defined as 300px in the CSS.</p>

```
</head>
<body>

<h2>Padding and element
width</h2>

<div class="ex1">This div is
300px wide.</div>
<br>

<div class="ex2">The width of
this div is 350px, even though
it is defined as 300px in the
CSS.</div>

</body>
</html>
```

6. CSS Height/Width

a, CSS Setting height and width

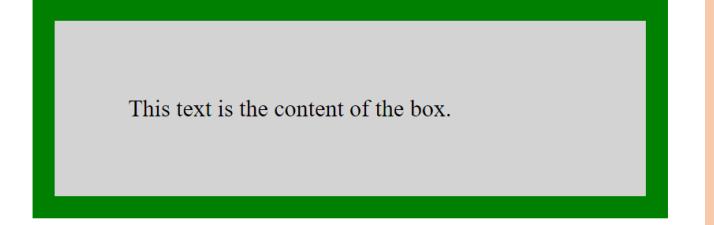
Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { height: 200px; width: 50%; background-color: powderblue; } </style> </head> <body> <h2>Set the height and width of an element</h2> <div>This div element has a height of 200px and a width of 50%.</div> </body> </html></pre>	<p>Set the height and width of an element</p> <p>This div element has a height of 200px and a width of 50%.</p>

b, Setting max-width

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { max-width: 500px; height: 100px; background-color: powderblue; } </style> </head> <body> <h2>Set the max-width of an element</h2> <div>This div element has a height of 100px and a max-width of 500px.</div> <p>Resize the browser window to see the effect.</p> </body> </html></pre>	<p>Set the max-width of an element</p> <p>This div element has a height of 100px and a max-width of 500px.</p> <p>Resize the browser window to see the effect.</p>

7. CSS Box Model

a, The CSS Box Model

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { background-color: lightgrey; width: 300px; border: 15px solid green; padding: 50px; margin: 20px; } </style> </head> <body></pre>	<p>Demonstrating the Box Model</p> <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>  <p>This text is the content of the box.</p>

```
<h2>Demonstrating the Box Model</h2>

<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box.</div>

</body>
</html>
```

b, Width and Height of an Element

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> div { width: 320px; height: 50px; padding: 10px; border: 5px solid gray; margin: 0; } </style> </head> <body> <h2>Calculate the total width:</h2> <div>The picture above is 350px wide. The total width of this element is also 350px. The total height of this element is 80px.</div> </body> </html></pre>	<p>Calculate the total width:</p>  <p>The picture above is 350px wide. The total width of this element is also 350px. The total height of this element is 80px.</p>

8. CSS Outline

8.1. Outline

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { border: 2px solid black; outline: #4CAF50 solid 10px; margin: auto; padding: 20px; text-align: center; } </style> </head> <body> <h2>CSS Outline</h2> <p>This element has a 2px black border and a green outline with a width of 10px.</p> </body> </html></pre>	<p>CSS Outline</p> <p>This element has a 2px black border and a green outline with a width of 10px.</p>

8.2. Outline width

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.ex1 { border: 1px solid black; outline-style: solid; outline-color: red; outline-width: thin; } p.ex2 { border: 1px solid black; outline-style: solid; outline-color: red; outline-width: medium; } p.ex3 {</pre>	<p>The outline-width Property</p> <p>A thin outline.</p> <p>A medium outline.</p> <p>A thick outline.</p> <p>A 4px thick outline.</p>

Thực tập Kiến trúc và Giao thức IoT

```
border: 1px solid black;
outline-style: solid;
outline-color: red;
outline-width: thick;
}

p.ex4 {
  border: 1px solid black;
  outline-style: solid;
  outline-color: red;
  outline-width: 4px;
}
</style>
</head>
<body>

<h2>The outline-width
Property</h2>

<p class="ex1">A thin
outline.</p>
<p class="ex2">A medium
outline.</p>
<p class="ex3">A thick
outline.</p>
<p class="ex4">A 4px thick
outline.</p>

</body>
</html>
```

8.3. Outline Color

a, CSS Outline Color

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.ex1 { border: 2px solid black; outline-style: solid; outline-color: red; } p.ex2 { border: 2px solid black; outline-style: dotted; outline-color: blue;</pre>	<p>The outline-color Property</p> <p>The outline-color property is used to set the color of the outline.</p> <p>A solid red outline.</p> <p>A dotted blue outline.</p> <p>An outset grey outline.</p>

```

}

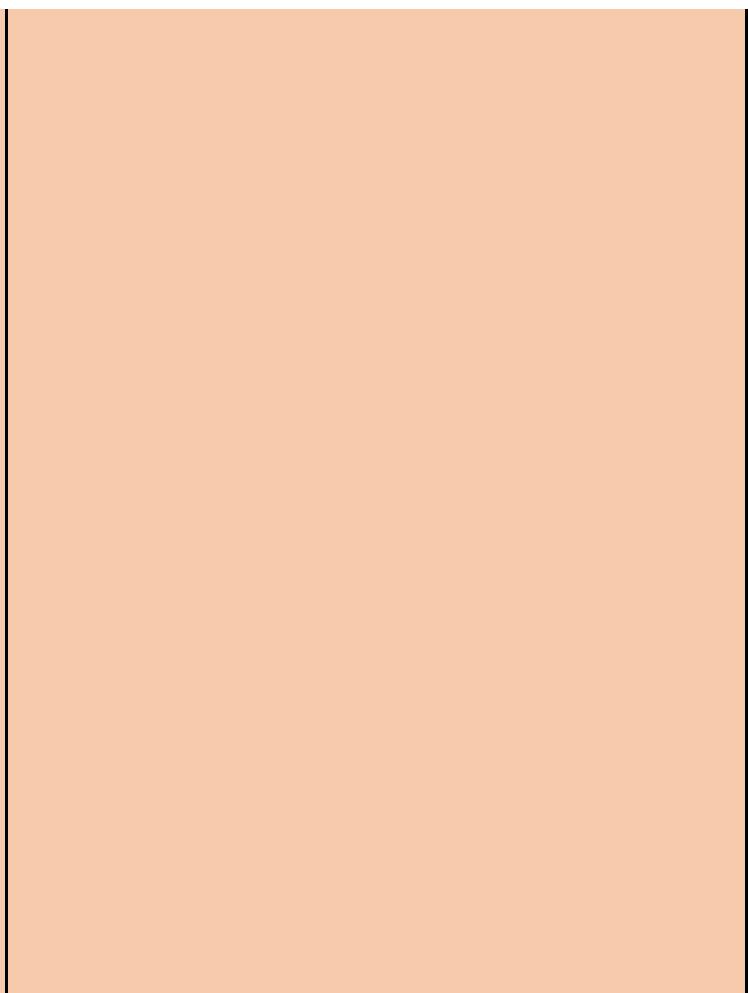
p.ex3 {
    border: 2px solid black;
    outline-style: outset;
    outline-color: grey;
}
</style>
</head>
<body>

<h2>The outline-color
Property</h2>
<p>The outline-color property
is used to set the color of the
outline.</p>

<p class="ex1">A solid red
outline.</p>
<p class="ex2">A dotted blue
outline.</p>
<p class="ex3">An outset grey
outline.</p>

</body>
</html>

```



b, HEX Values

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> p.ex1 { border: 2px solid black; outline-style: solid; outline-color: #ff0000; /* red */ } p.ex2 { border: 2px solid black; outline-style: dotted; outline-color: #0000ff; /* blue */ } p.ex3 { </pre>	<p>The outline-color Property</p> <p>The color of the outline can also be specified using a hexadecimal value (HEX):</p> <p>A solid red outline.</p> <p>A dotted blue outline.</p> <p>A solid grey outline.</p>

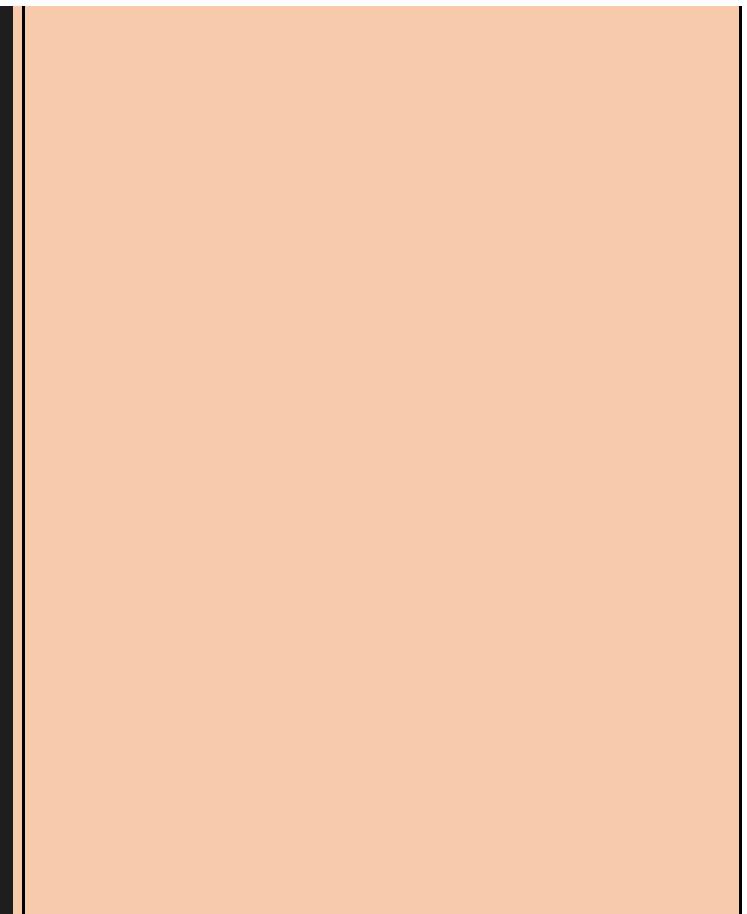
Thực tập Kiến trúc và Giao thức IoT

```
border: 2px solid black;
outline-style: solid;
outline-color: #bbbbbb; /* grey */
}
</style>
</head>
<body>

<h2>The outline-color Property</h2>
<p>The color of the outline can also be specified using a hexadecimal value (HEX):</p>

<p class="ex1">A solid red outline.</p>
<p class="ex2">A dotted blue outline.</p>
<p class="ex3">A solid grey outline.</p>

</body>
</html>
```



c, RGB Values

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.ex1 { border: 2px solid black; outline-style: solid; outline-color: rgb(255, 0, 0); /* red */ } p.ex2 { border: 2px solid black; outline-style: dotted; outline-color: rgb(0, 0, 255); /* blue */ } p.ex3 { border: 2px solid black; outline-style: solid;</pre>	<h3>The outline-color Property</h3> <p>The color of the outline can also be specified using RGB values:</p> <p>A solid red outline.</p> <p>A dotted blue outline.</p> <p>A solid grey outline.</p>

```

outline-color: rgb(187, 187,
187); /* grey */
}
</style>
</head>
<body>

<h2>The outline-color
Property</h2>
<p>The color of the outline can
also be specified using RGB
values:</p>

<p class="ex1">A solid red
outline.</p>
<p class="ex2">A dotted blue
outline.</p>
<p class="ex3">A solid grey
outline.</p>

</body>
</html>

```

d, HSL Values

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> p.ex1 { border: 2px solid black; outline-style: solid; outline-color: hsl(0, 100%, 50%); /* red */ } p.ex2 { border: 2px solid black; outline-style: dotted; outline-color: hsl(240, 100%, 50%); /* blue */ } p.ex3 { border: 2px solid black; outline-style: solid; outline-color: hsl(0, 0%, 73%); /* grey */ } </pre>	<p>The outline-color Property</p> <p>The color of the outline can also be specified using HSL values:</p> <p>A solid red outline.</p> <p>A dotted blue outline.</p> <p>A solid grey outline.</p>

```
}

</style>
</head>
<body>

<h2>The outline-color
Property</h2>
<p>The color of the outline can
also be specified using HSL
values:</p>

<p class="ex1">A solid red
outline.</p>
<p class="ex2">A dotted blue
outline.</p>
<p class="ex3">A solid grey
outline.</p>

</body>
</html>
```

8.4. Outline Shorthand

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.ex1 {outline: dashed;} p.ex2 {outline: dotted red;} p.ex3 {outline: 5px solid yellow;} p.ex4 {outline: thick ridge pink;} </style> </head> <body> <h2>The outline Property</h2> <p class="ex1">A dashed outline.</p> <p class="ex2">A dotted red outline.</p> <p class="ex3">A 5px solid yellow outline.</p> <p class="ex4">A thick ridge pink outline.</p></pre>	<p>The outline Property</p> <p>A dashed outline.</p> <p>A dotted red outline.</p> <p>A 5px solid yellow outline.</p> <p>A thick ridge pink outline.</p>

```
</body>  
</html>
```

8.5. Outline Offset

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { margin: 30px; border: 1px solid black; outline: 1px solid red; outline-offset: 15px; } </style> </head> <body> <h2>The outline-offset Property</h2> <p>This paragraph has an outline 15px outside the border edge.</p> </body> </html></pre>	<p>The outline-offset Property</p> <div style="border: 1px solid red; padding: 5px; width: fit-content;"><p>This paragraph has an outline 15px outside the border edge.</p></div>

9. CSS Text

9.1. Text color

a, Text Color

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { color: blue; } h1 { color: green; } </style> </head> <body> <h1>This is heading 1</h1> <p>This is text.</p> </body></pre>	<p>This is heading 1</p> <p>This is text.</p>

```
</html>
```

b, Text Color and Background Color

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-color: lightgrey; color: blue; } h1 { background-color: black; color: white; } div { background-color: blue; color: white; } </style> </head> <body> <h1>This is a Heading</h1> <p>This page has a grey background color and a blue text.</p> <div>This is a div.</div> </body> </html></pre>	<p>This is a Heading</p> <p>This page has a grey background color and a blue text.</p> <p>This is a div.</p>

9.2. Text Alignment

a, Text Alignment

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-align: center; } h2 { text-align: left; } h3 { text-align: right; } </style></pre>	<p>Heading 1 (center)</p> <p>Heading 2 (left)</p> <p>Heading 3 (right)</p> <p>The three headings above are aligned center, left and right.</p>

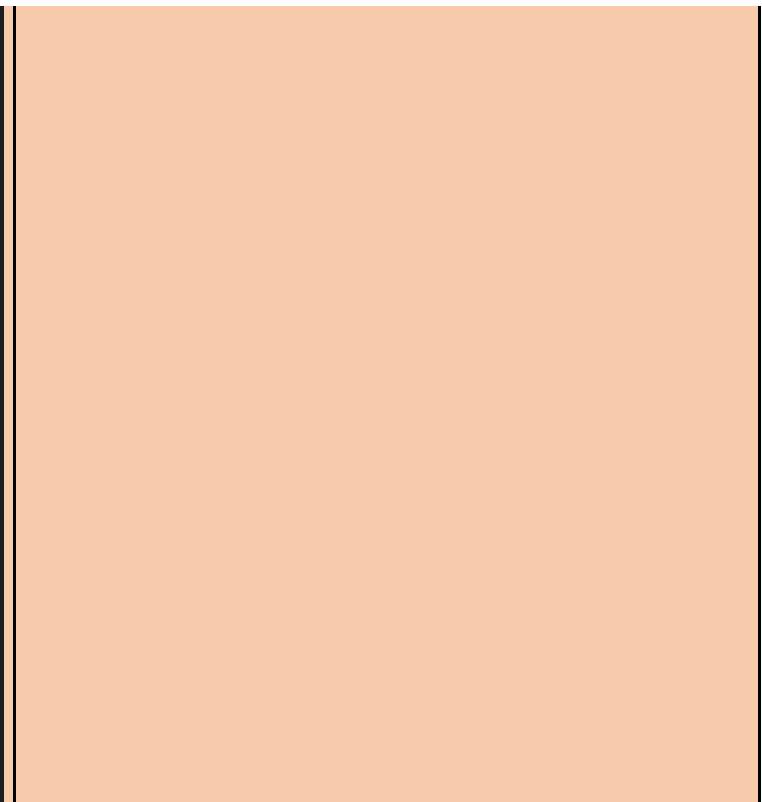
```
</head>
<body>
<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>
<p>The three headings above are
aligned center, left and
right.</p>
</body>
</html>
```

b, Text Align Last

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.a { text-align-last: right; } p.b { text-align-last: center; } p.c { text-align-last: justify; } </style> </head> <body> <h1>The text-align-last Property</h1> <h2>text-align-last: right:</h2> <p class="a">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p> <h2>text-align-last: center:</h2></pre>	<p>The text-align-last Property</p> <p>text-align-last: right:</p> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p> <p>text-align-last: center:</p> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p> <p>text-align-last: justify:</p> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>

Thực tập Kiến trúc và Giao thức IoT

```
<p class="b">Lorem ipsum dolor  
sit amet, consectetur  
adipiscing elit. Etiam semper  
diam at erat pulvinar, at  
pulvinar felis blandit.  
Vestibulum volutpat tellus  
diam, consequat gravida libero  
rhoncus ut.</p>  
  
<h2>text-align-last:  
justify:</h2>  
<p class="c">Lorem ipsum dolor  
sit amet, consectetur  
adipiscing elit. Etiam semper  
diam at erat pulvinar, at  
pulvinar felis blandit.  
Vestibulum volutpat tellus  
diam, consequat gravida libero  
rhoncus ut.</p>  
</body>  
</html>
```



c, Text Direction

Code	Result
<!DOCTYPE html> <html> <head> <style> p.ex1 { direction: rtl; unicode-bidi: bidi-override; } </style> </head> <body> <p>This is the default text direction.</p> <p class="ex1">This is right- to-left text direction.</p> </body> </html>	This is the default text direction. .noitcerid txet tfel-ot-thgir si sihT

d, Vertical Alignment

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
img.a {
    vertical-align: baseline;
}

img.b {
    vertical-align: text-top;
}

img.c {
    vertical-align: text-bottom;
}

img.d {
    vertical-align: sub;
}

img.e {
    vertical-align: super;
}
</style>
</head>
<body>

<h1>The vertical-align Property</h1>

<h2>vertical-align: baseline
(default):</h2>
<p>An  image with a default
alignment.</p>

<h2>vertical-align: text-top:</h2>
<p>An  image with a text-top
alignment.</p>

<h2>vertical-align: text-bottom:</h2>
<p>An  image with a text-bottom
alignment.</p>

<h2>vertical-align: sub:</h2>
```

The vertical-align Property

vertical-align: baseline (default):

An  image with a default alignment.

vertical-align: text-top:

An  image with a text-top alignment.

vertical-align: text-bottom:

An  image with a text-bottom alignment.

vertical-align: sub:

An  image with a sub alignment.

vertical-align: sup:

An  image with a super alignment.

```
<p>An  image with a sub alignment.</p>

<h2>vertical-align: sup:</h2>
<p>An  image with a super alignment.</p>

</body>
</html>
```

9.3. Text Decoration

a, Thêm Decoration line

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-decoration: overline; } h2 { text-decoration: line-through; } h3 { text-decoration: underline; } p.ex { text-decoration: overline underline; } </style> </head> <body> <h1>Overline text decoration</h1> <h2>Line-through text decoration</h2> <h3>Underline text decoration</h3> <p class="ex">Overline and underline text decoration.</p></pre>	<p>Overline text decoration</p> <p>Line-through text decoration</p> <p><u>Underline text decoration</u></p> <p><u>Overline and underline text decoration.</u></p> <p>Note: It is not recommended to underline text that is not a link, as this often confuses the reader.</p>

```
<p><strong>Note:</strong> It is  
not recommended to underline text  
that is not a link, as this often  
confuses  
the reader.</p>  
  
</body>  
</html>
```

b, Specify a Color for the Decoration Line

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-decoration-line: overline; text-decoration-color: red; } h2 { text-decoration-line: line- through; text-decoration-color: blue; } h3 { text-decoration-line: underline; text-decoration-color: green; } p { text-decoration-line: overline underline; text-decoration-color: purple; } </style> </head> <body> <h1>Overline text decoration</h1> <h2>Line-through text decoration</h2> <h3>Underline text decoration</h3> <p>Overline and underline text decoration.</p> </body></pre>	<p>Overline text decoration</p> <p>Line-through text decoration</p> <p>Underline text decoration</p> <p>Overline and underline text decoration.</p>

</html>

c, Specify a Style for the Decoration Line

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-decoration-line: underline; text-decoration-style: solid; /* this is default */ } h2 { text-decoration-line: underline; text-decoration-style: double; }</pre>	<h1><u>Heading 1</u></h1>
<pre>h3 { text-decoration-line: underline; text-decoration-style: dotted; } p.ex1 { text-decoration-line: underline; text-decoration-style: dashed; }</pre>	<h2><u>Heading 2</u></h2>
<pre>p.ex2 { text-decoration-line: underline; text-decoration-style: wavy; }</pre>	<h3><u>Heading 3</u></h3>
<pre>p.ex3 { text-decoration-line: underline; text-decoration-color: red; text-decoration-style: wavy; } </style> </head> <body> <h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <p class="ex1">A paragraph.</p></pre>	<p>A <u>paragraph</u>.</p> <p><u>Another paragraph</u>.</p> <p><u>Another paragraph</u>.</p>

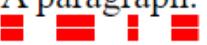
```
<p class="ex2">Another
paragraph.</p>
<p class="ex3">Another
paragraph.</p>

</body>
</html>
```

d, Specify the Thickness for the Decoration Line

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-decoration-line: underline; text-decoration-thickness: auto; /* this is default */ } h2 { text-decoration-line: underline; text-decoration-thickness: 5px; } h3 { text-decoration-line: underline; text-decoration-thickness: 25%; } p { text-decoration-line: underline; text-decoration-color: red; text-decoration-style: double; text-decoration-thickness: 5px; } </style> </head> <body> <h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <p>A paragraph.</p> </body> </html></pre>	<h1><u>Heading 1</u></h1> <h2><u>Heading 2</u></h2> <h3><u>Heading 3</u></h3> <p>A paragraph.</p> 

e, The Shorthand Property

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-decoration: underline; } h2 { text-decoration: underline red; } h3 { text-decoration: underline red double; } p { text-decoration: underline red double 5px; } </style> </head> <body> <h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <p>A paragraph.</p> </body> </html></pre>	<h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <p>A paragraph.</p> 

f, A Small Tip

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> a { text-decoration: none; } </style> </head></pre>	<h2>Using text-decoration: none</h2> <p>A link with no underline: W3Schools.com</p>

```
<body>

<h1>Using text-decoration:
none</h1>

<p>A link with no underline:
<a
href="https://www.w3schools.co
m">W3Schools.com</a></p>

</body>
</html>
```

9.4 Text Transformation

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.uppercase { text-transform: uppercase; } p.lowercase { text-transform: lowercase; } p.capitalize { text-transform: capitalize; } </style> </head> <body> <h1>Using the text-transform property</h1> <p class="uppercase">This text is transformed to uppercase.</p> <p class="lowercase">This text is transformed to lowercase.</p> <p class="capitalize">This text is capitalized.</p> </body> </html></pre>	<p>Using the text-transform property</p> <p>THIS TEXT IS TRANSFORMED TO UPPERCASE.</p> <p>this text is transformed to lowercase.</p> <p>This Text Is Capitalized.</p>

9.5 Text Spacing

a, Text Indentation

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { text-indent: 50px; } </style> </head> <body> <h1>Using text-indent</h1> <p>text here.</p> </body> </html></pre>	<p>Using text-indent</p> <p>text here.</p>

b, Letter Spacing

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h2 { letter-spacing: 5px; } h3 { letter-spacing: -2px; } </style> </head> <body> <h1>Using letter-spacing</h1> <h2>This is heading 1</h2> <h3>This is heading 2</h3> </body> </html></pre>	<p>Using letter-spacing</p> <p>This is heading 1</p> <p>This is heading 2</p>

c, Line Height

GVHD: Trương Quang Phúc

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.small { line-height: 0.7; } p.big { line-height: 1.8; } </style> </head> <body> <h1>Using line-height</h1> <p> This is a paragraph with a standard line-height.
 The default line height in most browsers is about 110% to 120%.
 </p> <p class="small"> This is a paragraph with a smaller line-height.
 This is a paragraph with a smaller line-height.
 </p> <p class="big"> This is a paragraph with a bigger line-height.
 This is a paragraph with a bigger line-height.
 </p> </body> </html></pre>	<h2>Using line-height</h2> <p>This is a paragraph with a standard line-height. The default line height in most browsers is about 110% to 120%.</p> <p>This is a paragraph with a smaller line-height. This is a paragraph with a smaller line-height.</p> <p>This is a paragraph with a bigger line-height. This is a paragraph with a bigger line-height.</p>

d, Word Spacing

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    word-spacing: 10px;
}

p.two {
    word-spacing: -2px;
}
</style>
</head>
<body>
<h1>Using word-spacing</h1>
<p>This is a paragraph with normal word spacing.</p>
<p class="one">This is a paragraph with larger word spacing.</p>
<p class="two">This is a paragraph with smaller word spacing.</p>
</body>
</html>
```

Using word-spacing

This is a paragraph with normal word spacing.

This is a paragraph with larger word spacing.

This is a paragraph with smaller word spacing.

e, White Space

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { white-space: nowrap; } </style> </head> <body> <h1>Using white-space</h1> <p> This is some text that will not wrap. </pre>	<h2>Using white-space</h2> <p>This is some text that will not wrap. This is some text that will not wrap.</p> <p>Try to remove the white-space property to see the difference!</p>

```
This is some text that will not
wrap.
</p>
<p>Try to remove the white-space
property to see the
difference!</p>
</body>
</html>
```

is some text that will not wrap. This is some text that will not

9.6 Text Shadow

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-shadow: 2px 2px; } </style> </head> <body> <h1>Text-shadow effect!</h1> </body> </html></pre>	<p>Text-shadow effect!</p>

10. CSS Fonts

10.1 Font Family

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .p1 { font-family: "Times New Roman", Times, serif; } </style> </head> <body> <p class="p1">CSS font-family</p> </body> </html></pre>	<p>CSS font-family</p> <p>This is a paragraph, shown in the Times New Roman font.</p> <p>This is a paragraph, shown in the Arial font.</p> <p>This is a paragraph, shown in the Lucida Console font.</p>

```
.p2 {  
    font-family: Arial, Helvetica,  
    sans-serif;  
}  
.p3 {  
    font-family: "Lucida Console",  
    "Courier New", monospace;  
}  
</style>  
</head>  
<body>  
<h1>CSS font-family</h1>  
<p class="p1">This is a  
paragraph, shown in the Times New  
Roman font.</p>  
<p class="p2">This is a  
paragraph, shown in the Arial  
font.</p>  
<p class="p3">This is a  
paragraph, shown in the Lucida  
Console font.</p>  
</body>  
</html>
```

10.2 Font Web safe

a, Fallback Fonts

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p { font-family: Tahoma, Verdana, sans- serif; } </style> </head> <body> <h1>CSS Fallback Fonts</h1> <p>This is a paragraph.</p> <p>This is another paragraph.</p> </body> </html></pre>	<p>CSS Fallback Fonts</p> <p>This is a paragraph.</p> <p>This is another paragraph.</p>

b, Best Web Safe Fonts for HTML and CSS

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

- Lưu ý: Trước khi bạn xuất bản trang web của mình, hãy luôn kiểm tra cách phông chữ của bạn xuất hiện trên các trình duyệt và thiết bị khác nhau và luôn sử dụng phông chữ dự phòng!

10.3 Font fallbacks

- Dưới đây là một số phông chữ dự phòng thường được sử dụng, được sắp xếp theo 5 họ phông chữ chung:

+ Serif Fonts

font-family	Ví dụ
"Times New Roman", Times, serif	<p>This is a Heading</p> <p>This is a paragraph.</p>
Georgia, serif	<p>This is a Heading</p> <p>This is a paragraph.</p>
Garamond, serif	<p>This is a Heading</p> <p>This is a paragraph.</p>

+ Sans-serif fonts

font-family	Ví dụ
Arial, Helvetica, sans-serif	<p>This is a Heading</p> <p>This is a paragraph.</p>

Tahoma, Verdana, sans-serif	This is a Heading This is a paragraph.
"Trebuchet MS", Helvetica, sans-serif	This is a Heading This is a paragraph.
Geneva, Verdana, sans-serif	This is a Heading This is a paragraph.

+ Monospace fonts

font-family	Ví dụ
"Courier New", Courier, monospace	This is a Heading This is a paragraph.

+ Cursive fonts

font-family	Ví dụ
"Brush Script MT", cursive	<i>This is a Heading</i> This is a paragraph.

+ Fantasy Fonts

font-family	Ví dụ
Copperplate, Papyrus, fantasy	<i>This is a Heading</i> This is a paragraph.

- Mẹo:Ngoài ra, hãy xem tất cả các Phòng chữ Google có sẵn và cách sử dụng chúng.

10.4 Fonts style

a, Font style

- Thuộc tính font-style chủ yếu được sử dụng để chỉ định văn bản in nghiêng.
- Thuộc tính này có ba giá trị:
 - + normal - Văn bản được hiển thị bình thường
 - + italic - Văn bản được hiển thị in nghiêng
 - + oblique - Văn bản "nghiêng" (xiên rất giống với chữ nghiêng, nhưng ít được hỗ trợ hơn).

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-style: normal;
}

p.italic {
  font-style: italic;
}

p.oblique {
  font-style: oblique;
}
</style>
</head>
<body>

<h1>The font-style property</h1>

<p class="normal">This is a
paragraph in normal style.</p>
<p class="italic">This is a
paragraph in italic style.</p>
<p class="oblique">This is a
paragraph in oblique style.</p>

</body>
</html>
```

The font-style property

This is a paragraph in normal style.

This is a paragraph in italic style.

This is a paragraph in oblique style.

b, Font weight

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.normal { font-weight: normal; } p.light { font-weight: lighter; } p.thick { font-weight: bold; } p.thicker { font-weight: 900; } </style> </head> <body> <h1>The font-weight property</h1> <p class="normal">This is a paragraph.</p> <p class="light">This is a paragraph.</p> <p class="thick">This is a paragraph.</p> <p class="thicker">This is a paragraph.</p> </body> </html></pre>	<p>The font-weight property</p> <p>This is a paragraph.</p> <p>This is a paragraph.</p> <p>This is a paragraph.</p> <p>This is a paragraph.</p>

c, Font Variant

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.normal { font-variant: normal; } p.small { font-variant: small-caps; }</pre>	<p>The font-variant property</p> <p>My name is Hege Refsnes.</p> <p>MY NAME IS HEGE REFSNES.</p>

```
    font-variant: small-caps;
}
</style>
</head>
<body>

<h1>The font-variant property</h1>

<p class="normal">My name is Hege
Refsnes.</p>
<p class="small">My name is Hege
Refsnes.</p>

</body>
</html>
```

10.5 Font Size

- Thuộc tính font-size thiết lập kích thước của văn bản.
- Khả năng quản lý kích thước văn bản rất quan trọng trong thiết kế web. Tuy nhiên, bạn không nên sử dụng chức năng điều chỉnh cỡ chữ để đoạn văn trông giống tiêu đề, hoặc tiêu đề giống đoạn văn.

a, Set font size with pixels

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { font-size: 40px; } h2 { font-size: 30px; } p { font-size: 14px; } </style> </head> <body> <h1>This is heading 1</h1> <h2>This is heading 2</h2> <p>This is a paragraph.</p> <p>This is another paragraph.</p></pre>	<p>This is heading 1</p> <p>This is heading 2</p> <p>This is a paragraph.</p> <p>This is another paragraph.</p>

```
</body>
</html>
```

b, Set font size with Em

- Để cho phép người dùng thay đổi kích thước văn bản , nhiều nhà phát triển sử dụng em thay vì pixel. $1\text{em} = 16 \text{ pixel}$.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { font-size: 2.5em; /* 40px/16=2.5em */ } h2 { font-size: 1.875em; /* 30px/16=1.875em */ } p { font-size: 0.875em; /* 14px/16=0.875em */ } </style> </head> <body> <h1>This is heading 1</h1> <h2>This is heading 2</h2> <p>This is a paragraph.</p> </body> </html></pre>	<p>This is heading 1</p> <p>This is heading 2</p> <p>This is a paragraph.</p>

c, Sử dụng kết hợp Phân trăm với em

- Giải pháp hoạt động trong tất cả các trình duyệt là đặt kích thước phông chữ mặc định theo phần trăm cho phần tử `<body>`:

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  font-size: 100%;
}

h1 {
  font-size: 2.5em;
}

h2 {
  font-size: 1.875em;
}

p {
  font-size: 0.875em;
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

This is heading 1

This is heading 2

This is a paragraph.

d, Response Font size

- Kích thước văn bản có thể được đặt bằng đơn vị vw, có nghĩa là "chiều rộng khung nhìn".
- Bằng cách đó, kích thước văn bản sẽ tuân theo kích thước của cửa sổ trình duyệt:

Code	Result
<pre><!DOCTYPE html> <html> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <body> <h1 style="font- size:10vw;">Responsive Text</h1> <p style="font-size:5vw;">Resize the browser window to see how the text size scales.</p> <p style="font-size:5vw;">Use the "vw" unit when sizing the</pre>	<p>Responsive Text</p> <p>Resize the browser window to see how the text size scales.</p> <p>Use the "vw" unit when sizing the text. 10vw will set the size to 10% of the viewport width.</p>

```
text. 10vw will set the size to  
10% of the viewport width.</p>  
</body>  
</html>
```

10.6 Google fonts

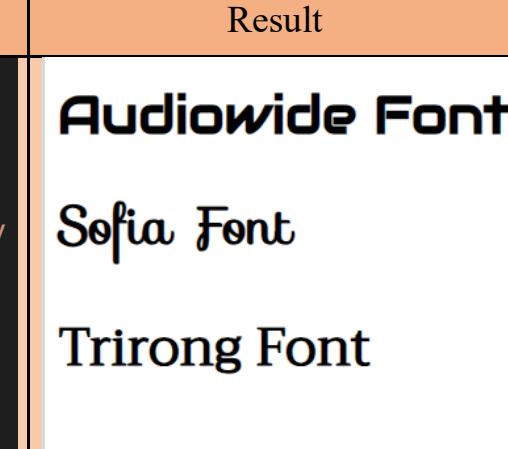
a, Cách dùng

- Chỉ cần thêm một liên kết định kiểu đặc biệt vào phần <head> rồi tham khảo phông chữ trong CSS.

Code	Result
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia"> <style> body { font-family: "Sofia", sans-serif; } </style> </head> <body> <h1>Sofia Font</h1> <p>Lorem ipsum dolor sit amet.</p> <p>123456790</p> </body> </html></pre>	

b, Sử dụng nhiều font của Google

- Để sử dụng nhiều phông chữ của Google, chỉ cần phân tách tên phông chữ bằng ký tự “|”, như sau:

Code	Result
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Audiowide Sofia Trirong"> <style> h1.a { font-family: "Audiowide", sans-serif; }</pre>	

Thực tập Kiến trúc và Giao thức IoT

```
h1.b {  
    font-family: "Sofia", sans-serif;  
}  
  
h1.c {  
    font-family: "Trirong", serif;  
}  
</style>  
</head>  
<body>  
  
<h1 class="a">audiowide Font</h1>  
  
<h1 class="b">Sofia Font</h1>  
  
<h1 class="c">Trirong Font</h1>  
  
</body>  
</html>
```

c, Tạo kiểu cho Google font

Code	Result
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia"> <style> body { font-family: "Sofia", sans-serif; font-size: 30px; text-shadow: 3px 3px 3px #ababab; } </style> </head> <body> <h1>Sofia Font</h1> <p>Lorem ipsum dolor sit amet.</p> <p>123456790</p> </body> </html></pre>	

d, Kích hoạt hiệu ứng phông chữ

- Google cũng đã kích hoạt các hiệu ứng phông chữ khác nhau mà bạn có thể sử dụng.

Thực tập Kiến trúc và Giao thức IoT

- Trước tiên, hãy thêm effect=effectname vào Google API, sau đó thêm tên lớp đặc biệt vào thành phần sẽ sử dụng hiệu ứng đặc biệt. Tên lớp luôn bắt đầu bằng font-effect- và kết thúc bằng effectname.

Code	Result
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia&effect=fire"> <style> body { font-family: "Sofia", sans-serif; font-size: 30px; } </style> </head> <body> <h1 class="font-effect-fire">Sofia on Fire</h1> <p class="font-effect-fire">Lorem ipsum dolor sit amet.</p> <p class="font-effect-fire">123456790</p> </body> </html></pre>	

10.7 Font Pairings

- Quy tắc:

+ Phân bù: Luôn an toàn khi tìm thấy các cặp phông chữ bổ sung cho nhau. Một sự kết hợp phông chữ tuyệt vời phải hài hòa, không quá giống hoặc quá khác nhau.

+ Sử dụng các siêu họ phông chữ: Siêu họ phông chữ là một tập hợp các phông chữ được thiết kế để hoạt động tốt với nhau. Vì vậy, sử dụng các phông chữ khác nhau trong cùng một siêu họ là an toàn.

Ví dụ: siêu họ Lucida chứa các phông chữ sau: Lucida Sans, Lucida Serif, Lucida Typewriter Sans, Lucida Typewriter Serif và Lucida Math.

+ Sự tương phản là vua: Hai phông chữ quá giống nhau thường sẽ xung đột. Tuy nhiên, sự tương phản, được thực hiện đúng cách, sẽ mang lại hiệu quả tốt nhất cho từng phông chữ.

Ví dụ: Kết hợp serif với sans serif là một sự kết hợp nổi tiếng.

Một siêu họ mạnh bao gồm cả các biến thể serif và sans serif của cùng một phông chữ (ví dụ: Lucida và Lucida Sans).

Thực tập Kiến trúc và Giao thức IoT

+ Chỉ chọn một chủ: Một phông chữ nên là chủ. Điều này thiết lập hệ thống phân cấp cho các phông chữ trên trang của bạn. Điều này có thể đạt được bằng cách thay đổi kích thước, trọng lượng và màu sắc.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-color: black; font-family: Verdana, sans-serif; font-size: 16px; color: gray; } h1 { font-family: Georgia, serif; font-size: 60px; color: white; } </style> </head> <body> <h1>Beautiful Norway</h1> <p>Text.</p> </body> </html></pre>	<p>Beautiful Norway</p> <p>Text.</p>

10.8 Font shorthand

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.a { font: 20px Arial, sans-serif; } p.b { font: italic small-caps bold 12px/30px Georgia, serif; } </style> </head> <body> <h1>The font Property</h1></pre>	<p>The font Property</p> <p>This is a paragraph.</p> <p>THIS IS A PARAGRAPH.</p>

```
<p class="a">This is a  
paragraph.</p>  
<p class="b">This is a  
paragraph.</p>  
</body>  
</html>
```

11. CSS Icons

- Các biểu tượng có thể dễ dàng được thêm vào trang HTML của bạn bằng cách sử dụng thư viện biểu tượng.

11.1. Font Awesome Icons

- Để sử dụng các biểu tượng Font Awesome, hãy truy cập fontawesome.com, đăng nhập và nhận mã để thêm vào phần <head> trên trang HTML của bạn:

```
<script src="https://kit.fontawesome.com/yourcode.js"  
crossorigin="anonymous"></script>
```

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>Font Awesome Icons</title> <meta name="viewport" content="width=device-width, initial-scale=1"> <script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script> <!--Get your own code at fontawesome.com--> </head> <body> <h1>Font Awesome icon library</h1> <p>Some Font Awesome icons:</p> <i class="fas fa-cloud"></i> <i class="fas fa-heart"></i> <i class="fas fa-car"></i> <i class="fas fa-file"></i> <i class="fas fa-bars"></i> <p>Styled Font Awesome icons (size and color):</p> <i class="fas fa-cloud" style="font-size:24px;"></i> <i class="fas fa-cloud" style="font-size:36px;"></i> <i class="fas fa-cloud" style="font-size:48px; color:red;"></i></pre>	<p>Font Awesome icon library</p> <p>Some Font Awesome icons:</p> <p>Cloud Heart Car File Bars</p> <p>Styled Font Awesome icons (size and color):</p> 

```
<i class="fas fa-cloud" style="font-size:60px;color:lightblue;"></i>  
  
</body>  
</html>
```

11.2. Bootstrap Icons

- Để sử dụngglyphicon Bootstrap, hãy thêm dòng sau vào phần <head> trên trang HTML của bạn:

```
<link rel = "stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"  
">
```

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>Bootstrap Icons</title> <meta name="viewport" content="width=device-width, initial-scale=1"> <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"> </head> <body class="container"> <h1>Bootstrap icon library</h1> <p>Some Bootstrap icons:</p> <i class="glyphicon glyphicon-cloud"></i> <i class="glyphicon glyphicon-remove"></i> <i class="glyphicon glyphicon-user"></i> <i class="glyphicon glyphicon-envelope"></i> <i class="glyphicon glyphicon-thumbs-up"></i>

 <p>Styled Bootstrap icons (size and color):</p> <i class="glyphicon glyphicon-cloud" style="font-size:24px;"></i> <i class="glyphicon glyphicon-cloud" style="font-size:36px;"></i> <i class="glyphicon glyphicon-cloud" style="font-size:48px;color:red;"></i> <i class="glyphicon glyphicon-cloud" style="font-size:60px;color:lightblue;"></i> </body> </html></pre>	<p>Bootstrap icon library</p> <p>Some Bootstrap icons:</p> <p>Some Bootstrap icons (size and color):</p> 

11.3. Google Icon

- Để sử dụng các biểu tượng Google, hãy thêm dòng sau vào bên trong phần <head> trên trang HTML của bạn:

```
<link rel="stylesheet"  
      href="https://fonts.googleapis.com/icon?family=Material+Icons">
```

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>Google Icons</title> <meta name="viewport" content="width=device-width, initial-scale=1"> <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons"> </head> <body> <h1>Google icon library</h1> <p>Some Google icons:</p> <i class="material-icons">cloud</i> <i class="material- icons">favorite</i> <i class="material- icons">attachment</i> <i class="material- icons">computer</i> <i class="material- icons">traffic</i>

 <p>Styled Google icons (size and color):</p> <i class="material-icons" style="font-size:24px;">cloud</i> <i class="material-icons" style="font-size:36px;">cloud</i> <i class="material-icons" style="font- size:48px;color:red;">cloud</i> <i class="material-icons" style="font- size:60px;color:lightblue;">cloud</i > </body> </html></pre>	<p>Google icon library</p> <p>Some Google icons:</p> <p></p> <p>Styled Google icons (size and color):</p> <p></p>

12. List

12.1 HTML Lists and CSS List Properties

- Trong HTML, có hai loại danh sách chính:

+ Danh sách không có thứ tự (``) - các mục trong danh sách được đánh dấu bằng dấu đầu dòng

+ Danh sách có thứ tự (``) - các mục trong danh sách được đánh dấu bằng số hoặc chữ cái

- Thuộc tính danh sách CSS cho phép bạn:

+ Đặt các điểm đánh dấu mục danh sách khác nhau cho danh sách được sắp xếp;

+ Đặt các điểm đánh dấu mục danh sách khác nhau cho danh sách không có thứ tự;

+ Đặt hình ảnh làm điểm đánh dấu mục danh sách;

+ Thêm màu nền vào danh sách và mục danh sách.

12.2 Các điểm đánh dấu mục danh sách khác nhau

- Thuộc tính `list-style-type` chỉ định loại điểm đánh dấu mục danh sách.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> ul.a { list-style-type: circle; } ul.b { list-style-type: square; } ol.c { list-style-type: upper-roman; } ol.d { list-style-type: lower-alpha; } </style> </head> <body> <h2>The list-style-type Property</h2> <p>Example of unordered lists:</p> <ul class="a"> Coffee Tea Coca Cola <ul class="b"> Coffee Tea Coca Cola </pre>	<p>The list-style-type Property</p> <p>Example of unordered lists:</p> <ul style="list-style-type: none"> ○ Coffee ○ Tea ○ Coca Cola <ul style="list-style-type: none"> ■ Coffee ■ Tea ■ Coca Cola <p>Example of ordered lists:</p> <ul style="list-style-type: none"> I. Coffee II. Tea III. Coca Cola <ul style="list-style-type: none"> a. Coffee b. Tea c. Coca Cola

Thực tập Kiến trúc và Giao thức IoT

```
</ul>
<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
</body>
</html>
```

12.3 Hình ảnh làm điểm đánh dấu mục danh sách

Code	Result
<!DOCTYPE html> <html> <head> <style> ul { list-style-image: url('sqpurple.gif'); } </style> </head> <body> <h2>The list-style-image Property</h2> <p>The list-style-image property specifies an image as the list item marker:</p> Coffee Tea Coca Cola </body> </html>	<p>The list-style-image Property</p> <p>The list-style-image property specifies an image as the list item marker:</p> <ul style="list-style-type: none">■ Coffee■ Tea■ Coca Cola

12.4 Định vị các điểm đánh dấu mục danh sách

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a {
    list-style-position: outside;
}
ul.b {
    list-style-position: inside;
}
</style>
</head>
<body>
<h1>The list-style-position
Property</h1>
<h2>list-style-position: outside
(default):</h2>
<ul class="a">
    <li>Coffee - A brewed drink
prepared from roasted coffee
beans, which are the seeds of
berries from the Coffea plant</li>
</ul>
<h2>list-style-position:
inside:</h2>
<ul class="b">
    <li>Coffee - A brewed drink
prepared from roasted coffee
beans, which are the seeds of
berries from the Coffea plant</li>
</ul>
</body>
</html>
```

The list-style-position Property

list-style-position: outside (default):

- Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant

list-style-position: inside:

- Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant

12.5 Xóa cài đặt mặc định

- Thuộc tính list-style-type:none cũng có thể được sử dụng để xóa đi điểm đánh dấu/dấu đầu dòng. Lưu ý rằng danh sách cũng có lè và phần đệm mặc định. Để loại bỏ điều này, hãy thêm lè:0 và phần đệm:0 vào hoặc :

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.demo {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
</style>
</head>
<body>

<p>Default list:</p>
<ul>
    <li>Coffee</li>
    <li>Tea</li>
    <li>Coca Cola</li>
</ul>

<p>Remove bullets, margin and
padding from list:</p>
<ul class="demo">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Coca Cola</li>
</ul>

</body>
</html>
```

Default list:

- Coffee
- Tea
- Coca Cola

Remove bullets, margin and padding from list:

Coffee
Tea
Coca Cola

12.6 List - Shorthand property

- Thuộc tính kiểu danh sách là thuộc tính viết tắt. Nó được sử dụng để đặt tất cả các thuộc tính danh sách trong một khai báo.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> ul { list-style: square inside url("sqpurple.gif"); } </style> </head> <body> <h2>The list-style Property</h2> </pre>	<h2>The list-style Property</h2> <ul style="list-style-type: none"> ■ Coffee ■ Tea ■ Coca Cola

```
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>
</html>
```

- Khi sử dụng shorthand property, thứ tự của các giá trị thuộc tính là:

+ list-style-type (nếu chỉ định list-style-image, giá trị của thuộc tính này sẽ được hiển thị nếu hình ảnh vì lý do nào đó không thể hiển thị);

+ list-style-position (xác định xem các điểm đánh dấu mục danh sách sẽ xuất hiện bên trong hay bên ngoài luồng nội dung);

+ list-style-image (chỉ định một hình ảnh làm điểm đánh dấu mục danh sách).

12.7 Styling list with colors

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> ol { background: #ff9999; padding: 20px; } ul { background: #3399ff; padding: 20px; } ol li { background: #ffe5e5; color: darkred; padding: 5px; margin-left: 35px; } ul li { background: #cce5ff; color: darkblue; margin: 5px; } </style> </head> <body> <h1>Styling Lists With Colors</h1> Coffee Tea Coca Cola </pre>	<p>Styling Lists With Colors</p>

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
</body>
</html>
```

13. CSS Tables

13.1 Table borders

a, Table borders

- Để chỉ định đường viền bảng trong CSS, hãy sử dụng thuộc tính border.
- Ví dụ bên dưới chỉ định đường viền liền nét cho các phần tử `<table>`, `<th>` và `<td>`.

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid; } </style> </head> <body> <h2>Add a border to a table:</h2> <table> <tr> <th>Firstname</th> <th>Lastname</th> </tr> <tr> <td>Peter</td> <td>Griffin</td> </tr> <tr> <td>Lois</td> <td>Griffin</td> </tr> </table> </body> </html></pre>	<p>Add a border to a table:</p> <table border="1"><thead><tr><th>Firstname</th><th>Lastname</th></tr></thead><tbody><tr><td>Peter</td><td>Griffin</td></tr><tr><td>Lois</td><td>Griffin</td></tr></tbody></table>	Firstname	Lastname	Peter	Griffin	Lois	Griffin
Firstname	Lastname						
Peter	Griffin						
Lois	Griffin						

b, Full-Width Table

Code	Result
------	--------

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid;
}

table {
  width: 100%;
}
</style>
</head>
<body>

<h2>Full-width Table</h2>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
  </tr>
</table>

</body>
</html>
```

Full-width Table

Firstname	Lastname
Peter	Griffin
Lois	Griffin

c, Collapse Table Borders

- Thuộc tính border-collapse thiết lập xem các đường viền của bảng có được thu gọn thành một đường viền duy nhất hay không:

Code	Result						
<!DOCTYPE html> <html> <head> <style> table, td, th { border: 1px solid; } </style> </head> <body> <h2>Let the table borders collapse</h2> <table> <tr> <th>Firstname</th> <th>Lastname</th> </tr> <tr> <td>Peter</td> <td>Griffin</td> </tr> <tr> <td>Lois</td> <td>Griffin</td> </tr> </table> </body> </html>	<h2>Let the table borders collapse</h2> <table border="1"><thead><tr><th>Firstname</th><th>Lastname</th></tr></thead><tbody><tr><td>Peter</td><td>Griffin</td></tr><tr><td>Lois</td><td>Griffin</td></tr></tbody></table>	Firstname	Lastname	Peter	Griffin	Lois	Griffin
Firstname	Lastname						
Peter	Griffin						
Lois	Griffin						

```

table {
  width: 100%;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Let the table borders
collapse</h2>






```

13.2. Table Size

- Chiều rộng và chiều cao của bảng được xác định bởi thuộc tính width và height.

Code	Result									
<pre> <!DOCTYPE html> <html> <head> <style> table, td, th { border: 1px solid black; } table { border-collapse: collapse; width: 100%; } th { height: 70px; } </style> </head> </pre>	<p>The width and height Properties</p> <p>Set the width of the table, and the height of the table header row:</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Savings</th></tr> </thead> <tbody> <tr> <td>Peter</td><td>Griffin</td><td>\$100</td></tr> <tr> <td>Lois</td><td>Griffin</td><td>\$150</td></tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100	Lois	Griffin	\$150
Firstname	Lastname	Savings								
Peter	Griffin	\$100								
Lois	Griffin	\$150								

```

<body>
<h2>The width and height
Properties</h2>
<p>Set the width of the
table, and the height of the
table header row:</p>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
</table>
</body>
</html>

```

13.3 Table Alignment

a, Horizontal Alignment

- Thuộc tính text-align đặt căn chỉnh theo chiều ngang (như trái, phải hoặc giữa) của nội dung trong <th> hoặc <td>.
- Theo mặc định, nội dung của phần tử <th> được căn giữa và nội dung của phần tử <td> được căn trái.
- Để căn giữa nội dung của các phần tử <td>, hãy sử dụng text-align: center;

Code	Result									
<pre> <!DOCTYPE html> <html> <head> <style> table, td, th { border: 1px solid black; } table { border-collapse: collapse; width: 100%; } td { text-align: center; } </pre>	<p>The text-align Property</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Savings</th></tr> </thead> <tbody> <tr> <td>Peter</td><td>Griffin</td><td>\$100</td></tr> <tr> <td>Lois</td><td>Griffin</td><td>\$150</td></tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100	Lois	Griffin	\$150
Firstname	Lastname	Savings								
Peter	Griffin	\$100								
Lois	Griffin	\$150								

```

}
</style>
</head>
<body>
<h2>The text-align Property</h2>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
</table>
</body>
</html>

```

b, Vertical Alignment

- Thuộc tính vertical-align đặt căn chỉnh theo chiều dọc (như trên, dưới hoặc giữa) của nội dung trong <th> hoặc <td>.
- Theo mặc định, căn chỉnh theo chiều dọc của nội dung trong bảng là ở giữa (đối với cả phần tử <th> và <td>).

Code	Result									
<pre> <!DOCTYPE html> <html> <head> <style> table, td, th { border: 1px solid black; } table { border-collapse: collapse; width: 100%; } td { height: 50px; vertical-align: bottom; } </style> </pre>	<p>The vertical-align Property</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Savings</th></tr> </thead> <tbody> <tr> <td>Peter</td><td>Griffin</td><td>\$100</td></tr> <tr> <td>Lois</td><td>Griffin</td><td>\$150</td></tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100	Lois	Griffin	\$150
Firstname	Lastname	Savings								
Peter	Griffin	\$100								
Lois	Griffin	\$150								

```
</head>
<body>
<h2>The vertical-align
Property</h2>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
</table>
</body>
</html>
```

13.4 Table Style

a, Table Padding

- Để kiểm soát khoảng cách giữa đường viền và nội dung trong bảng, hãy sử dụng thuộc tính `padding` trên các phần tử `<td>` và `<th>`.

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table, td, th { border: 1px solid #ddd; text-align: left; } table { border-collapse: collapse; width: 100%; } th, td { padding: 15px; } </style> </head> <body></pre>	<p>The padding Property</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Savings</th></tr> </thead> <tbody> <tr> <td>Peter</td><td>Griffin</td><td>\$100</td></tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100
Firstname	Lastname	Savings					
Peter	Griffin	\$100					

Thực tập Kiến trúc và Giao thức IoT

```
<h2>The padding Property</h2>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
</table>
</body>
</html>
```

b, Horizontal dividers

- Thêm thuộc tính border-bottom vào <th> và <td> cho cải phân cách ngang.

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table { border-collapse: collapse; width: 100%; } th, td { padding: 8px; text-align: left; border-bottom: 1px solid #ddd; } </style> </head> <body> <h2>Bordered Table Dividers</h2> <table> <tr> <th>Firstname</th> <th>Lastname</th> <th>Savings</th> </tr> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> </table> </body> </html></pre>	<p>Bordered Table Dividers</p> <table><thead><tr><th>Firstname</th><th>Lastname</th><th>Savings</th></tr></thead><tbody><tr><td>Peter</td><td>Griffin</td><td>\$100</td></tr></tbody></table>	Firstname	Lastname	Savings	Peter	Griffin	\$100
Firstname	Lastname	Savings					
Peter	Griffin	\$100					

Thực tập Kiến trúc và Giao thức IoT

```
<td>$100</td>
</tr>
</table>
</body>
</html>
```

c, Hoverable Table

- Sử dụng bộ chọn :hover trên <tr> để đánh dấu các hàng trong bảng khi di chuột qua.

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table { border-collapse: collapse; width: 100%; } th, td { padding: 8px; text-align: left; border-bottom: 1px solid #ddd; } tr:hover {background- color: coral;} </style> </head> <body> <h2>Hoverable Table</h2> <table> <tr> <th>First Name</th> <th>Last Name</th> <th>Points</th> </tr> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> </table> </body> </html></pre>	<p>Hoverable Table</p> <table><thead><tr><th>First Name</th><th>Last Name</th><th>Points</th></tr></thead><tbody><tr><td>Peter</td><td>Griffin</td><td>\$100</td></tr></tbody></table>	First Name	Last Name	Points	Peter	Griffin	\$100
First Name	Last Name	Points					
Peter	Griffin	\$100					

d, Striped Table

Thực tập Kiến trúc và Giao thức IoT

- Đối với các bảng sọc ngựa vằn, hãy sử dụng bộ chọn nth-child() và thêm màu nền cho tất cả các hàng trong bảng chẵn (hoặc lẻ).

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table { border-collapse: collapse; width: 100%; } th, td { text-align: left; padding: 8px; } tr:nth-child(even) {background-color: #f2f2f2;} </style> </head> <body> <h2>Striped Table</h2> <table> <tr> <th>First Name</th> <th>Last Name</th> <th>Points</th> </tr> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> </table> </body> </html></pre>	<p>Striped Table</p> <table> <thead> <tr> <th>First Name</th> <th>Last Name</th> <th>Points</th> </tr> </thead> <tbody> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> </tbody> </table>	First Name	Last Name	Points	Peter	Griffin	\$100
First Name	Last Name	Points					
Peter	Griffin	\$100					

e, Table Color

Code	Result						
<pre><!DOCTYPE html> <html> <head> <style> table { border-collapse: collapse; width: 100%; } th, td { text-align: left; }</pre>	<p>Colored Table Header</p> <table> <thead> <tr style="background-color: green;"> <th>Firstname</th> <th>Lastname</th> <th>Savings</th> </tr> </thead> <tbody> <tr> <td>Peter</td> <td>Griffin</td> <td>\$100</td> </tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100
Firstname	Lastname	Savings					
Peter	Griffin	\$100					

Thực tập Kiến trúc và Giao thức IoT

```
padding: 8px;
}
tr:nth-child(even){background-
color: #f2f2f2}
th {
  background-color: #04AA6D;
  color: white;
}
</style>
</head>
<body>
<h2>Colored Table Header</h2>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
</table>
</body>
</html>
```

13.5 Table Responsive

- Bảng phản hồi sẽ hiển thị thanh cuộn ngang nếu màn hình quá nhỏ để hiển thị đầy đủ nội dung.

Code	Result																				
<pre><!DOCTYPE html> <html> <head> <style> table { border-collapse: collapse; width: 100%; } th, td { text-align: left; padding: 8px; } tr:nth-child(even) {background- color: #f2f2f2;} </style> </head> <body></pre>	<p>Responsive Table</p> <table><thead><tr><th>First Name</th><th>Last Name</th><th>Points</th><th>Points</th><th>Points</th><th>Points</th><th>Points</th><th>Points</th><th>Points</th><th>Points</th></tr></thead><tbody><tr><td>Jill</td><td>Smith</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td></tr></tbody></table>	First Name	Last Name	Points	Jill	Smith	50	50	50	50	50	50	50	50							
First Name	Last Name	Points	Points	Points	Points	Points	Points	Points	Points												
Jill	Smith	50	50	50	50	50	50	50	50												

```
<h2>Responsive Table</h2>
<div style="overflow-x: auto;">
  <table>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
    </tr>
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
    </tr>
  </table>
</div>
</body>
</html>
```

14. CSS Navigation Bar

- Việc điều hướng dễ sử dụng là điều quan trọng đối với bất kỳ trang web nào.
- Với CSS, bạn có thể biến các menu HTML nhàn chán thành các thanh điều hướng đẹp mắt

- Thanh điều hướng cần có HTML tiêu chuẩn làm cơ sở.

14.1. Vertical Navigation Bar

a, Vertical Navigation Bar

Code	Result
------	--------

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
li a {
    display: block;
    width: 60px;
    background-color: #dddddd;
}
</style>
</head>
<body>
<ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#news">News</a></li>
    <li><a href="#contact">Contact</a></li>
    <li><a href="#about">About</a></li>
</ul>
<p>A background color is added to the links to show the link area.</p>
<p>Notice that the whole link area is clickable, not just the text.</p>
</body>
</html>
```

[Home](#)
[News](#)
[Contact](#)
[About](#)

A background color is added to the links to show the link area.

Notice that the whole link area is clickable, not just the text.

b, Active/Current Navigation Link

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 200px;
    background-color: #f1f1f1;
}
li a {
    display: block;
    color: #000;
    padding: 8px 16px;
    text-decoration: none;
}
li a.active {
    background-color: #04AA6D;
    color: white;
}
li a:hover:not(.active) {
    background-color: #555;
    color: white;
}
</style>
</head>
<body>
<h2>Vertical Navigation Bar</h2>
<ul>
    <li><a class="active"
    href="#home">Home</a></li>
    <li><a href="#news">News</a></li>
    <li><a
    href="#contact">Contact</a></li>
    <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Vertical Navigation Bar

Home

News

Contact

About

c, Center link & add border

- Thêm text-align:center vào hoặc <a> để căn giữa các liên kết.
- Thêm thuộc tính đường viền đế thêm đường viền xung quanh thanh điều hướng.
- Nếu bạn cũng muốn có đường viền bên trong thanh điều hướng, hãy thêm border-bottom vào tất cả các phần tử , ngoại trừ phần tử cuối cùng:

Code	Result
------	--------

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 200px;
    background-color: #f1f1f1;
    border: 1px solid #555;
}
li a {
    display: block;
    color: #000;
    padding: 8px 16px;
    text-decoration: none;
}
li {
    text-align: center;
    border-bottom: 1px solid #555;
}
li:last-child {
    border-bottom: none;
}
li a.active {
    background-color: #04AA6D;
    color: white;
}
li a:hover:not(.active) {
    background-color: #555;
    color: white;
}
</style>
</head>
<body>
<h2>Vertical Navigation Bar</h2>
<ul>
    <li><a class="active"
        href="#home">Home</a></li>
    <li><a
        href="#news">News</a></li>
    <li><a
        href="#contact">Contact</a></li>
    <li><a
        href="#about">About</a></li>
</ul>
</body>
</html>
```

Vertical Navigation Bar



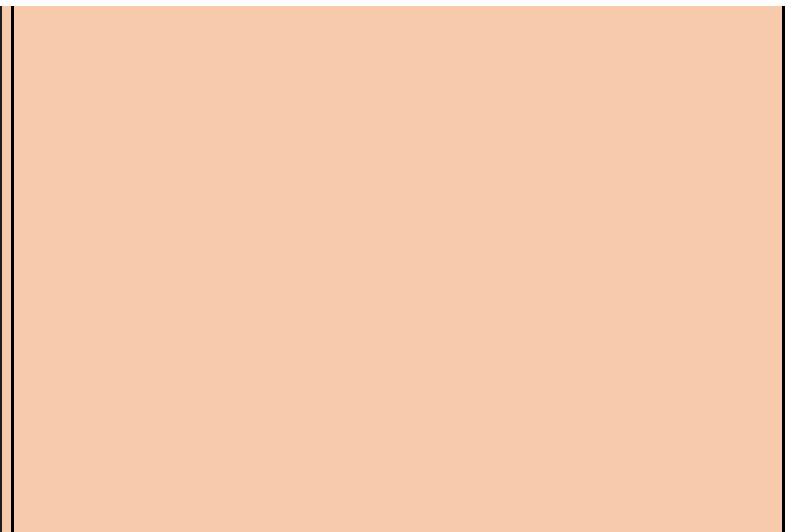
d, Full-height Fixed Vertical Navbar

GVHD: Trương Quang Phúc

Thực tập Kiến trúc và Giao thức IoT

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> body { margin: 0; } ul { list-style-type: none; margin: 0; padding: 0; width: 25%; background-color: #f1f1f1; position: fixed; height: 100%; overflow: auto; } li a { display: block; color: #000; padding: 8px 16px; text-decoration: none; } li a.active { background-color: #04AA6D; color: white; } li a:hover:not(.active) { background-color: #555; color: white; } </style> </head> <body> Home News Contact About <div style="margin- left:25%;padding:1px 16px;height:1000px;"></pre>	<p>Home News Contact About</p> <p>Some text.. Some text.. Some text.. Some text.. Some text.. Some text.. Some text..</p> <p>Fixed Full-height Side Nav</p> <p>Try to scroll this area, and see how the sidebar sticks to the page</p>
	<p>Home News Contact About</p> <p>Some text.. Some text.. Some text.. Some text.. Some text..</p>

```
<h2>Fixed Full-height Side  
Nav</h2>  
<h3>Try to scroll this  
area, and see how the sidenav  
sticks to the page</h3>  
<p>Some text..</p>  
<p>Some text..</p>  
<p>Some text..</p>  
<p>Some text..</p>  
<p>Some text..</p>  
<p>Some text..</p>  
</div>  
</body>  
</html>
```



14.2. Horizontal Navigation Bar

a, Inline List Items

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; } li { display: inline; } </style> </head> <body> Home News Contact About </body> </html></pre>	Home News Contact About

b, Floating list Items

Thực tập Kiến trúc và Giao thức IoT

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; } li { float: left; } li a { display: block; padding: 8px; background-color: #dddddd; } </style> </head> <body> Home News Contact About <p>Note: If a !DOCTYPE is not specified, floating items can produce unexpected results.</p> <p>Note: overflow:hidden is added to the ul element to prevent li elements from going outside of the list.</p> </body> </html></pre>	<p>Note: If a !DOCTYPE is not specified, floating items can produce unexpected results.</p> <p>Note: overflow:hidden is added to the ul element to prevent li elements from going outside of the list.</p>

c, Active/Current Navigation Bar

- Thêm một lớp "hoạt động" vào liên kết hiện tại để cho người dùng biết họ đang ở trang nào:

Code	Result
------	--------

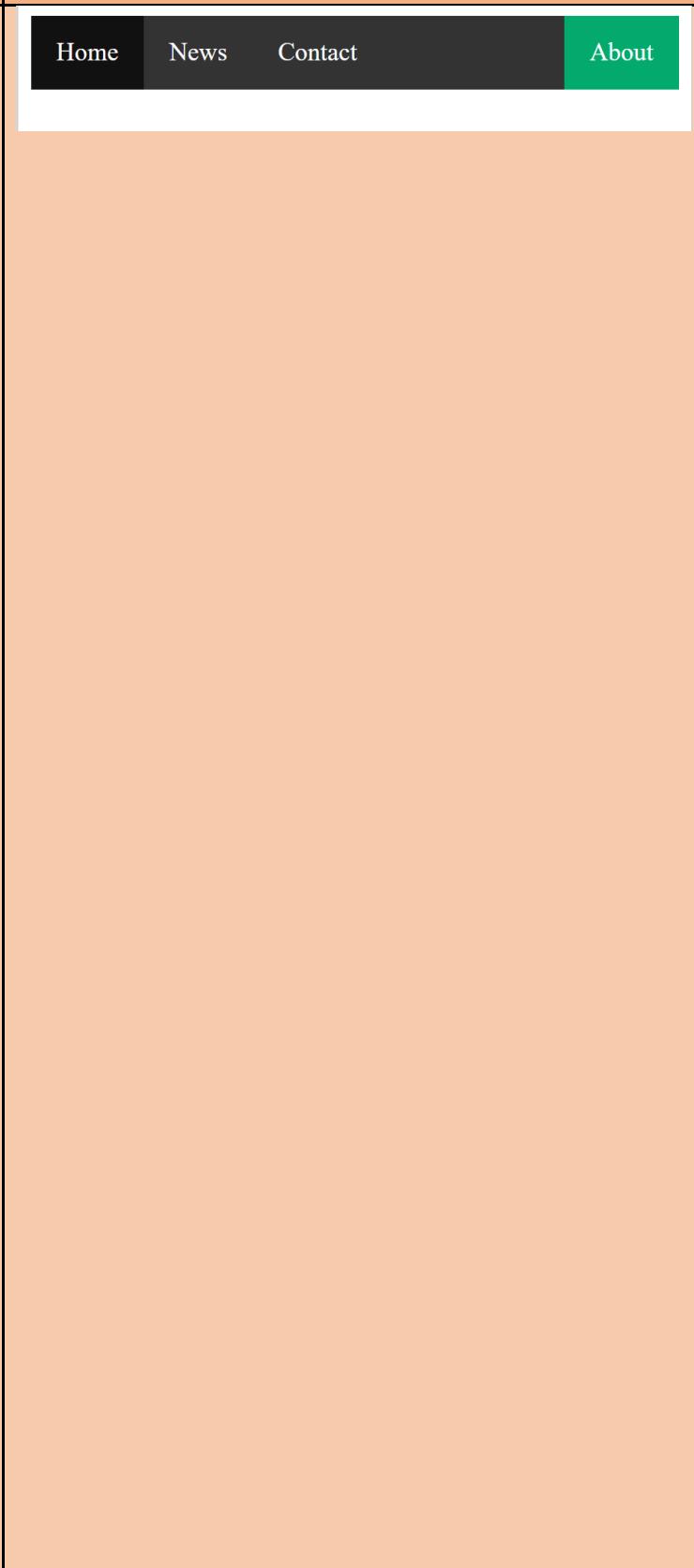
Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
}
li {
    float: left;
}
li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}
li a:hover:not(.active) {
    background-color: #111;
}
.active {
    background-color:
#04AA6D;
}
</style>
</head>
<body>
<ul>
    <li><a class="active"
href="#home">Home</a></li>
    <li><a
href="#news">News</a></li>
    <li><a
href="#contact">Contact</a>
</li>
    <li><a
href="#about">About</a></li>
>
</ul>
</body>
</html>
```

Home News Contact About

d, Right - Align link

- Căn phải các liên kết bằng cách thả nổi các mục danh sách sang bên phải (float:right;):

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; } li { float: left; } li a { display: block; color: white; text-align: center; padding: 14px 16px; text-decoration: none; } li a:hover:not(.active) { background-color: #111; } .active { background-color: #04AA6D; } </style> </head> <body> Home News Contact > <li style="float:right">About </body> </html> </pre>	

e, Border dividers

- Thêm thuộc tính border-right vào để tạo bộ chia liên kết:

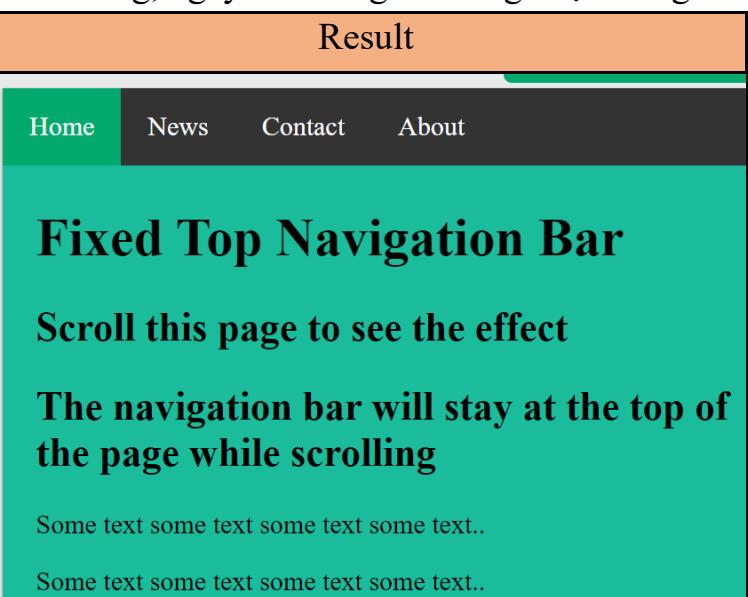
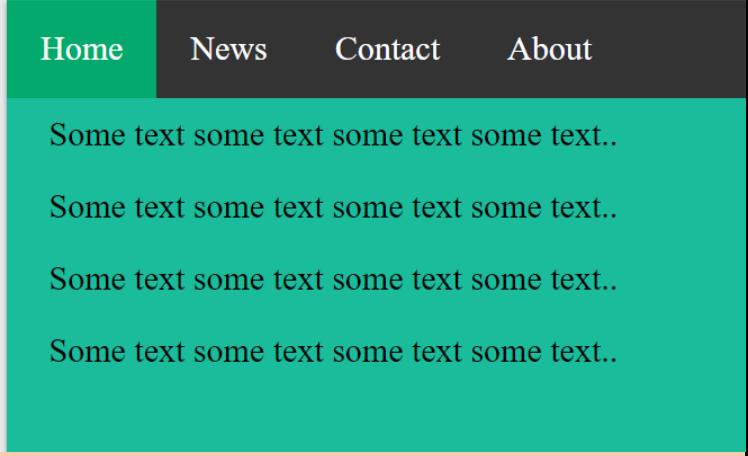
Thực tập Kiến trúc và Giao thức IoT

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; } li { float: left; border-right:1px solid #bbb; } li:last-child { border-right: none; } li a { display: block; color: white; text-align: center; padding: 14px 16px; text-decoration: none; } li a:hover:not(.active) { background-color: #111; } .active { background-color: #04AA6D; } </style> </head> <body> Home News Contact <li style="float:right">About </body> </html></pre>	

Thực tập Kiến trúc và Giao thức IoT

f, Fixed Navigation Bar

- Đặt thanh điều hướng ở đầu hoặc cuối trang, ngay cả khi người dùng cuộn trang:

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body {margin:0;} ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; position: fixed; top: 0; width: 100%; } li { float: left; } li a { display: block; color: white; text-align: center; padding: 14px 16px; text-decoration: none; } li a:hover:not(.active) { background-color: #111; } .active { background-color: #04AA6D; } </style> </head> <body> Home News Contact About</pre>	 <p>Home News Contact About</p> <h2>Fixed Top Navigation Bar</h2> <p>Scroll this page to see the effect</p> <p>The navigation bar will stay at the top of the page while scrolling</p> <p>Some text some text some text some text..</p> <p>Some text some text some text some text..</p>
	 <p>Home News Contact About</p> <p>Some text some text some text some text..</p>

Thực tập Kiến trúc và Giao thức IoT

```
</ul>
<div
style="padding:20px;margin-
top:30px;background-
color:#1abc9c;height:1500px;">
  <h1>Fixed Top Navigation
Bar</h1>
  <h2>Scroll this page to see
the effect</h2>
  <h2>The navigation bar will
stay at the top of the page
while scrolling</h2>
  <p>Some text some text some
text some text...</p>
  <p>Some text some text some
text some text...</p>
</div>
</body>
</html>
```



Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body {margin:0;} ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; position: fixed; bottom: 0; width: 100%; } li { float: left; } li a { display: block; }</pre>	<p>Fixed Bottom Navigation Bar</p> <p>Scroll this page to see the effect</p> <p>The navigation bar will stay at the bottom of the page while scrolling</p> <p>Some text some text some text some text..</p> <p>Some text some text some text some text</p> <p>Home News Contact About</p>

Thực tập Kiến trúc và Giao thức IoT

```
color: white;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}
li a:hover:not(.active) {
  background-color: #111;
}
.active {
  background-color: #04AA6D;
}
</style>
</head>
<body>
<ul>
  <li><a class="active"
href="#home">Home</a></li>
  <li><a
href="#news">News</a></li>
  <li><a
href="#contact">Contact</a></li>
>
  <li><a
href="#about">About</a></li>
</ul>
<div
style="padding:20px;background-
color:#1abc9c;height:1500px;">
  <h1>Fixed Bottom Navigation
Bar</h1>
  <h2>Scroll this page to see
the effect</h2>
  <h2>The navigation bar will
stay at the bottom of the page
while scrolling</h2>
  <p>Some text some text some
text some text..</p>
  <p>Some text some text some
text some text..</p>
  <p>Some text some text some
text some text..</p>
</div>
</body>
</html>
```

The screenshot shows a web page with a fixed bottom navigation bar. The navigation bar is a dark grey horizontal bar located at the bottom of the page. It contains four items: "Home" (which is highlighted in green), "News", "Contact", and "About". Above the navigation bar, there is a teal-colored header section with some placeholder text ("Some text some text some text..."). Below the header, the main content area has a light orange background. The content includes a heading "Fixed Bottom Navigation Bar", a sub-heading "Scroll this page to see the effect", and a paragraph of placeholder text.

g, Gray Navigation Bar

Code

Result

Thực tập Kiến trúc và Giao thức IoT

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    border: 1px solid #e7e7e7;
    background-color: #f3f3f3;
}
li {
    float: left;
}
li a {
    display: block;
    color: #666;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}
li a:hover:not(.active) {
    background-color: #ddd;
}
li a.active {
    color: white;
    background-color: #04AA6D;
}
</style>
</head>
<body>
<ul>
    <li><a class="active"
href="#home">Home</a></li>
    <li><a
href="#news">News</a></li>
    <li><a
href="#contact">Contact</a></li>
    >
    <li><a
href="#about">About</a></li>
</ul>
</body>
</html>
```



h, Sticky NavBar

- Thêm vị trí: dính; đến `` để tạo thanh điều hướng cố định.

Thực tập Kiến trúc và Giao thức IoT

- Phần tử cố định chuyển đổi giữa tương đối và cố định, tùy thuộc vào vị trí cuộn. Nó được định vị tương đối cho đến khi gặp một vị trí bù nhất định trong khung nhìn - sau đó nó "cố định" tại chỗ (như vị trí: cố định).

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { font-size: 28px; } ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; position: -webkit-sticky; /* Safari */ position: sticky; top: 0; } li { float: left; } li a { display: block; color: white; text-align: center; padding: 14px 16px; text-decoration: none; } li a:hover { background-color: #111; } .active { background-color: #4CAF50; } </style> </head> <body> <div class="header"></pre>	<p>Scroll Down</p> <p>Scroll down to see the sticky effect.</p> <p>Home News Contact</p> <p>Sticky Navigation Bar Example</p> <p>The navbar will stick to the top when you reach its scroll position.</p> <p>Note: Internet Explorer do not support sticky positioning and Safari requires a -webkit- prefix.</p> <p>texxt</p> <p>Home News Contact</p> <p>Sticky Navigation Bar Example</p> <p>The navbar will stick to the top when you reach its scroll position.</p> <p>Note: Internet Explorer do not support sticky positioning and Safari requires a -webkit- prefix.</p> <p>texxt</p> <p>texxt</p> <p>texxt</p> <p>texxt</p>

Thực tập Kiến trúc và Giao thức IoT

```
<h2>Scroll Down</h2>
<p>Scroll down to see the
sticky effect.</p>
</div>

<ul>
  <li><a class="active"
href="#home">Home</a></li>
  <li><a
href="#news">News</a></li>
  <li><a
href="#contact">Contact</a></li>
</ul>

<h3>Sticky Navigation Bar
Example</h3>
<p>The navbar will
<strong>stick</strong> to the
top when you reach its scroll
position.</p>
<p><strong>Note:</strong>
Internet Explorer do not
support sticky positioning and
Safari requires a -webkit-
prefix.</p>
<p>texxt</p>
<p>texxt</p>
<p>texxt</p>
<p>texxt</p>
</body>
</html>
```

15. CSS Image Gallery

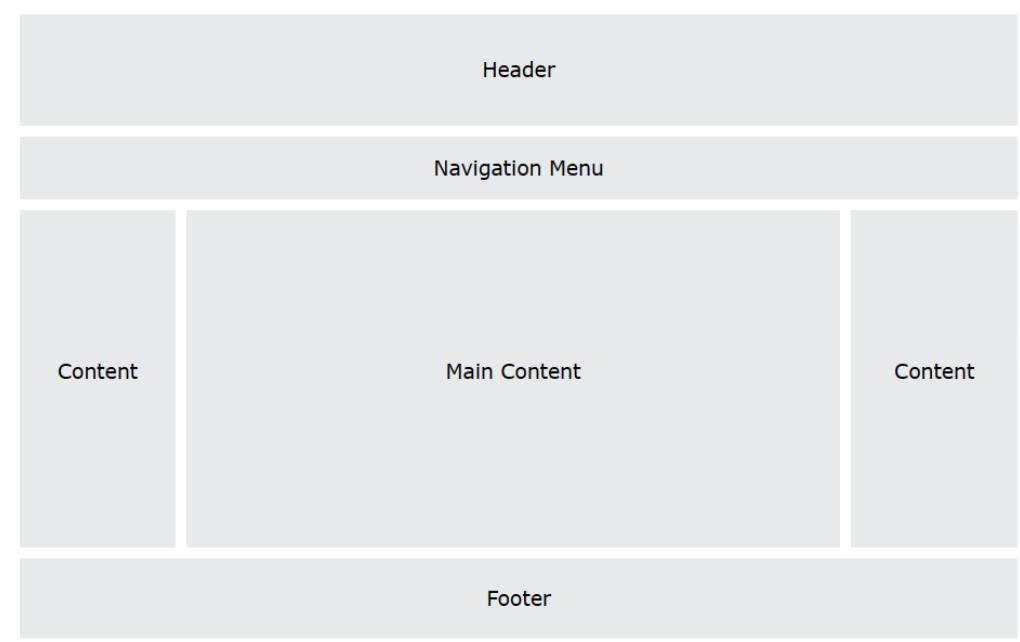
Code	Result	
<pre><!DOCTYPE html> <html> <head> <style> div.gallery { margin: 5px; border: 1px solid #ccc; float: left; width: 180px; } div.gallery:hover { border: 1px solid #777; } div.gallery img {</pre>	 Add a description of the image here	 Add a description of the image here

```
width: 100%;  
height: auto;  
}  
div.desc {  
    padding: 15px;  
    text-align: center;  
}  
</style>  
</head>  
<body>  
  
<div class="gallery">  
    <a target="_blank"  
href="img_5terre.jpg">  
          
    </a>  
    <div class="desc">Add a  
description of the image  
here</div>  
</div>  
  
<div class="gallery">  
    <a target="_blank"  
href="img_forest.jpg">  
          
    </a>  
    <div class="desc">Add a  
description of the image  
here</div>  
</div>  
</body>  
</html>
```

16. CSS Website Layout

16.1. Website Layout

- Bộ cục của một trang web thường được chia thành headers, menus, content và footer.



16.2. Header

- Header thường được đặt ở đầu của trang web (hoặc ở phía dưới bên phải một top navigation menu). Nó thường bao gồm một logo hoặc tên của website

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <title>CSS Website Layout</title> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial- scale=1"> <style> body { margin: 0; } .header { background-color: #f1f1f1; padding: 20px; text-align: center; } </style> </head> <body> <div class="header"> <h1>Header</h1> </div> </body> </html></pre>	<p>Header</p>

16.3. Navigation Bar

- Một Navigation Bar bao gồm một danh sách các đường dẫn giúp người truy cập chuyển hướng đến khắp trang web.

Code	Result
<pre> <!DOCTYPE html> <html lang="en"> <head> <title>CSS Website Layout</title> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <style> * { box-sizing: border-box; } body { margin: 0; } /* Style the header */ .header { background-color: #f1f1f1; padding: 20px; text-align: center; } /* Style the top navigation bar */ .topnav { overflow: hidden; background-color: #333; } /* Style the topnav links */ .topnav a { float: left; display: block; color: #f2f2f2; text-align: center; padding: 14px 16px; text-decoration: none; } /* Change color on hover */ .topnav a:hover { background-color: #ddd; color: black; } </style> </head> <body></pre>	<p style="text-align: center;">Header</p>  <p>The screenshot shows a dark header bar with the word "Header" centered above it. Below the header, there is a dark navigation bar containing three white text links labeled "Link".</p>

```
<div class="header">
  <h1>Header</h1>
</div>
<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>
</body>
</html>
```

16.4. Content

- Bộ cục của phần này phụ thuộc vào người dùng mà chúng ta muốn hướng tới. Phổ biến nhất là một hoặc một số các cột như dưới đây:
 - + 1-column (thường được sử dụng cho trình duyệt mobile);
 - + 2-column (thường được sử dụng cho máy tính bảng hay laptop);
 - + 3-column (thường được sử dụng cho desktops);



Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <title>CSS Website Layout</title> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"></pre>	<p>Header</p> <p>Resize the browser window to see the responsive effect.</p> <p>Link Link Link</p> <p>Column Column Column</p>

Thực tập Kiến trúc và Giao thức IoT

```
<style>
* {
  box-sizing: border-box;
}
body {
  margin: 0;
}
/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}
/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}
/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
/* Create three equal columns that
floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 15px;
}
/* Clear floats after the columns
*/
.row::after {
  content: "";
  display: table;
  clear: both;
}
/* Responsive layout - makes the
three columns stack on top of each
other instead of next to each
other */

```

```
@media screen and (max-width:600px) {  
    .column {  
        width: 100%;  
    }  
}  
</style>  
</head>  
<body>  
<div class="header">  
    <h1>Header</h1>  
    <p>Resize the browser window to  
see the responsive effect.</p>  
</div>  
<div class="topnav">  
    <a href="#">Link</a>  
    <a href="#">Link</a>  
    <a href="#">Link</a>  
</div>  
<div class="row">  
    <div class="column">  
        <h2>Column</h2>  
    </div>  
  
    <div class="column">  
        <h2>Column</h2>  
    </div>  
  
    <div class="column">  
        <h2>Column</h2>  
    </div>  
</div>  
</body>  
</html>
```

16.5. Unequal Columns

- Nội dung chính là lớn nhất và là phần quan trọng nhất trong trang web của bạn. Nên các cột với độ rộng không cân bằng nhau là phổ biến. Do đó phần lớn không gian được để dành cho nội dung chính. Các mảng còn lại của nội dung được sử dụng như một phương án thay thế hoặc nội dung liên qua tới main content. Thay đổi độ rộng với giá trị bạn thích, nhớ rằng tổng độ rộng chỉ có thể đạt đến 100%.

Code	Result
------	--------

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website
Layout</title>
<meta charset="utf-8">
<meta name="viewport"
content="width=device-
width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}
body {
  margin: 0;
}
/* Style the header */
.header {
  background-color:
#f1f1f1;
  padding: 20px;
  text-align: center;
}
/* Style the top
navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}
/* Style the topnav links
*/
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
/* Change color on hover
*/
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
/* Create three unequal
columns that floats next
to each other */
.column {
  float: left;
```

Header

Link Link Link

Side

Main Content

Side

Thực tập Kiến trúc và Giao thức IoT

```
padding: 10px;
}
/* Left and right column */
.column.side {
  width: 25%;
}
/* Middle column */
.column.middle {
  width: 50%;
}
/* Clear floats after the
columns */
.row::after {
  content: "";
  display: table;
  clear: both;
}
/* Responsive layout -
makes the three columns
stack on top of each
other instead of next to
each other */
@media screen and (max-
width: 600px) {
  .column.side,
  .column.middle {
    width: 100%;
  }
}
</style>
</head>
<body>
<div class="header">
  <h1>Header</h1>
</div>
<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>
<div class="row">
  <div class="column
side">
    <h2>Side</h2>
  </div>
  <div class="column
middle">
    <h2>Main Content</h2>
  </div>
```

```
<div class="column side">
  <h2>Side</h2>
</div>
</body>
</html>
```

16.6. Footer

- Footer được đặt ở cuối của trang web, nó thường bao gồm thông tin như là bản quyền và thông tin liên hệ.

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <title>CSS Website Layout</title> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <style> * { box-sizing: border-box; } body { margin: 0; } /* Style the header */ .header { background-color: #f1f1f1; padding: 20px; text-align: center; } /* Style the top navigation bar */ .topnav { overflow: hidden; background-color: #333; } /* Style the topnav links */ .topnav a { float: left; display: block; color: #f2f2f2; text-align: center; }</pre>	<p>Header</p> <p>The screenshot shows a website layout with the following structure:</p> <ul style="list-style-type: none">Header: A dark header bar containing three links labeled "Link".Main Content: A central white area with the text "Main Content" centered.Side: Two smaller white areas labeled "Side" on either side of the main content.Footer: A light gray footer area at the bottom of the page.

Thực tập Kiến trúc và Giao thức IoT

```
padding: 14px 16px;
text-decoration: none;
}
/* Change color on hover */
.topnav a:hover {
background-color: #ddd;
color: black;
}
/* Create three unequal
columns that floats next to
each other */
.column {
float: left;
padding: 10px;
}
/* Left and right column */
.column.side {
width: 25%;
}
/* Middle column */
.column.middle {
width: 50%;
}
/* Clear floats after the
columns */
.row::after {
content: "";
display: table;
clear: both;
}
/* Responsive layout -
makes the three columns
stack on top of each other
instead of next to each
other */
@media screen and (max-
width: 600px) {
.column.side,
.column.middle {
width: 100%;
}
}
/* Style the footer */
.footer {
background-color:
#f1f1f1;
padding: 10px;
text-align: center;
}
</style>
```

```
</head>
<body>
<div class="header">
  <h1>Header</h1>
</div>
<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>
<div class="row">
  <div class="column side">
    <h2>Side</h2>
  </div>
  <div class="column
middle">
    <h2>Main Content</h2>
  </div>
  <div class="column side">
    <h2>Side</h2>
  </div>
</div>
<div class="footer">
  <p>Footer</p>
</div>
</body>
</html>
```

16.6. Responsive Website Layout

- Bằng cách sử dụng các đoạn code CSS như trên, có thể tạo ra một trang web hoàn chỉnh và có khả năng tương tác một cách linh hoạt.

Code

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

body {
  font-family: Arial;
  padding: 10px;
  background: #f1f1f1;
}

/* Header/Blog Title */
.header {
```

Thực tập Kiến trúc và Giao thức IoT

```
padding: 30px;
text-align: center;
background: white;
}

.header h1 {
  font-size: 50px;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create two unequal columns that floats next to each other */
/* Left column */
.leftcolumn {
  float: left;
  width: 75%;
}

/* Right column */
.rightcolumn {
  float: left;
  width: 25%;
  background-color: #f1f1f1;
  padding-left: 20px;
}

/* Fake image */
.fakeimg {
  background-color: #aaa;
  width: 100%;
```

```
padding: 20px;
}

/* Add a card effect for articles */
.card {
    background-color: white;
    padding: 20px;
    margin-top: 20px;
}

/* Clear floats after the columns */
.row::after {
    content: "";
    display: table;
    clear: both;
}

/* Footer */
.footer {
    padding: 20px;
    text-align: center;
    background: #ddd;
    margin-top: 20px;
}

/* Responsive layout - when the screen is less than 800px wide, make the
two columns stack on top of each other instead of next to each other */
@media screen and (max-width: 800px) {
    .leftcolumn, .rightcolumn {
        width: 100%;
        padding: 0;
    }
}

/* Responsive layout - when the screen is less than 400px wide, make the
navigation links stack on top of each other instead of next to each other */
@media screen and (max-width: 400px) {
    .topnav a {
        float: none;
        width: 100%;
    }
}

</style>
</head>
<body>

<div class="header">
    <h1>My Website</h1>
    <p>Resize the browser window to see the effect.</p>

```

Thực tập Kiến trúc và Giao thức IoT

```
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" style="float:right">Link</a>
</div>

<div class="row">
  <div class="leftcolumn">
    <div class="card">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Dec 7, 2017</h5>
      <div class="fakeimg" style="height:200px;">Image</div>
      <p>Some text..</p>
    </div>
    <div class="card">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Sep 2, 2017</h5>
      <div class="fakeimg" style="height:200px;">Image</div>
      <p>Some text..</p>
    </div>
  </div>
  <div class="rightcolumn">
    <div class="card">
      <h2>About Me</h2>
      <div class="fakeimg" style="height:100px;">Image</div>
      <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
    </div>
    <div class="card">
      <h3>Popular Post</h3>
      <div class="fakeimg"><p>Image</p></div>
      <div class="fakeimg"><p>Image</p></div>
      <div class="fakeimg"><p>Image</p></div>
    </div>
    <div class="card">
      <h3>Follow Me</h3>
      <p>Some text..</p>
    </div>
  </div>
</div>

<div class="footer">
  <h2>Footer</h2>
</div>

</body>
</html>
```

Result

My Website

Resize the browser window to see the effect.

Link Link Link Link

TITLE HEADING
Title description, Dec 7, 2017
Image
Some text..

TITLE HEADING
Title description, Sep 2, 2017
Image
Some text..

About Me
Image
Some text about me in culpa qui officia deserunt mollit anim..

Popular Post
Image
Image
Image

Follow Me
Some text..

Footer

17. CSS Rounded Corners

- Với thuộc tính border-radius của CSS, bạn có thể đặt bất cứ phần tử nào với góc tròn

17.1. CSS *border-radius* Property

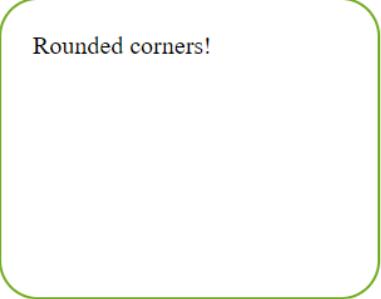
Code

Result

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners2 {
    border-radius: 25px;
    border: 2px solid #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
</style>
</head>
<body>
<h1>The border-radius Property</h1>
<p>Rounded corners for an element
with a border:</p>
<p id="rcorners2">Rounded
corners!</p>
</body>
</html>
```

The border-radius Property

Rounded corners for an element with a border:



Rounded corners!

17.2. CSS border-radius - Specify Each Corner

- Thuộc tính border-radius có thể có từ một đến bốn giá trị.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #rcorners1 { border-radius: 15px 50px 30px 5px; background: #73AD21; padding: 20px; width: 200px; height: 150px; } </style> </head> <body> <h1>The border-radius Property</h1> <p>Four values - border-radius: 15px 50px 30px 5px:</p> <p id="rcorners1"></p> </body> </html></pre>	<p>The border-radius Property</p> <p>Four values - border-radius: 15px 50px 30px 5px:</p> 

18. CSS Backgrounds

18.1. CSS Multiple Backgrounds

- CSS cho phép bạn thêm nhiều ảnh nền vào một phần tử, thông qua thuộc tính background-image.

- Các ảnh nền khác nhau được phân cách bằng dấu phẩy, và các ảnh được xếp bên trên của mỗi ảnh, nơi mà ảnh đầu tiên gần nhất với người dùng.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #example1 { background-image: url(img_flwr.gif), url(paper.gif); background-position: right bottom, left top; background-repeat: no-repeat, repeat; padding: 15px; } </style> </head> <body> <h1>Multiple Backgrounds</h1> <p>The following div element has two background images:</p> <div id="example1"> <h1>Lorem Ipsum Dolor</h1> </div> </body> </html></pre>	<p>Multiple Backgrounds</p> <p>The following div element has two background images:</p>

18.2. CSS Background Size

- Thuộc tính background-size của CSS cho phép bạn chỉ định kích thước của ảnh nền

- Kích thước đó có thể được chỉ định bằng chiều dài, phần trăm, hoặc sử dụng một trong hai keyword: contain hoặc cover.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #example1 { border: 1px solid black; background: url(img_flwr.gif); background-size: 100px 80px; }</pre>	<p>The background-size Property</p> <p>Resized background-image:</p>

```
background-repeat: no-repeat;
padding: 15px;
}
</style>
</head>
<body>
<h1>The background-size
Property</h1>
<p>Resized background-image:</p>
<div id="example1">
  <h2>Lorem Ipsum Dolor</h2>
</div>
</body>
</html>
```

18.3. Define Sizes of Multiple Background Images

- Thuộc tính background-size cũng chấp nhận nhiều giá trị cho background-size (sử dụng một dấu phẩy để phân cách danh sách), khi làm việc với đa ảnh nền.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #example1 { background: url(img_tree.gif) left top no-repeat, url(img_flwr.gif) right bottom no- repeat, url(paper.gif) left top repeat; padding: 15px; background-size: 50px, 130px, auto; } </style> </head> <body> <div id="example1"> <h1>Lorem Ipsum Dolor</h1> </div> </body> </html></pre>	

18.4. Full Size Background Image

- Nếu bạn muốn có một ảnh nền trên trang web luôn che phủ toàn bộ cửa sổ trình duyệt.

- Yêu cầu đó sẽ được thực hiện bằng các bước như sau:

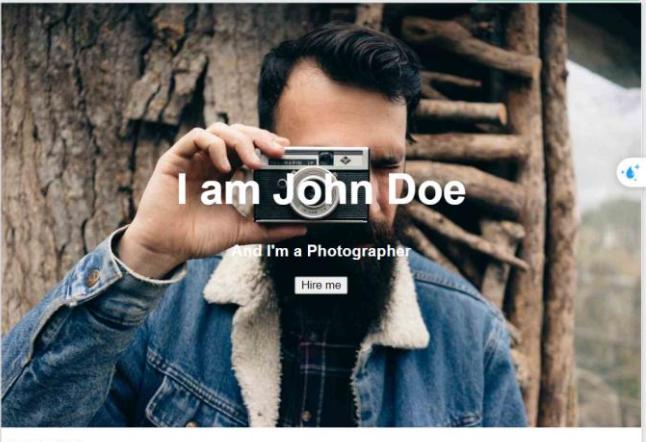
Thực tập Kiến trúc và Giao thức IoT

- + Điều toàn bộ trang web với ảnh (với không khoảng trắng)
- + Điều chỉnh kích thước ảnh cần dùng
- + Căn giữa ảnh trên trang web
- + Không bị ảnh hưởng bởi thanh cuộn

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> html { background: url(img_man.jpg) no-repeat center fixed; background-size: cover; } body { color: white; } </style> </head> <body> <h1>Full Page Background Image</h1> </body> </html></pre>	 <p>Full Page Background Image</p>

18.5. Hero Image

- Bạn cũng có thể sử dụng thuộc tính khác của background trên một thẻ <div> để tạo một hero image (một ảnh lớn với text), và đặt nó ở bất cứ nơi nào bạn muốn

Code	Result
<pre><!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width, initial-scale=1"> <style> body { margin: 0; font-family: Arial, Helvetica, sans-serif; } .hero-image { background: url(img_man.jpg) no-repeat center; background-size: cover; height: 500px; position: relative; }</pre>	 <p>I am John Doe</p> <p>And I'm a Photographer</p> <p>Hire me</p> <p>Page content...</p>

```

}
.hero-text {
    text-align: center;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    color: white;
}
</style>
</head>
<body>
<div class="hero-image">
    <div class="hero-text">
        <h1 style="font-size:50px">I
am John Doe</h1>
        <h3>And I'm a
Photographer</h3>
        <button>Hire me</button>
    </div>
</div>
<p>Page content...</p>
</body>
</html>

```

18.6. CSS background-origin Property

- Thuộc tính background-origin chỉ định nơi ảnh nền được định vị.
- Thuộc tính này có 3 giá trị khác nhau:
 - + Border-box - ảnh nền bắt đầu từ góc phía trên của đường viền
 - + Padding-box (mặc định) ảnh nền bắt đầu từ góc phía trên của viền padding
 - + Content-box - ảnh nền bắt đầu từ góc phía trên của nội dung

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> #example1 { border: 10px solid black; padding: 35px; background: url(img_flwr.gif); background-repeat: no-repeat; } </style> </head> <body> <h1>The background-origin Property</h1> </pre>	<p>The background-origin Property</p> <p>No background-origin (padding-box is default):</p>

```
<p>No background-origin (padding-box is default):</p>
<div id="example1">
  <h2>Lorem Ipsum Dolor</h2>
</div>
</body>
</html>
```

18.7. CSS *background-clip* Property

- Thuộc tính *background-clip* của CSS chỉ định vùng vẽ của nền.
- Có ba giá trị khác nhau cho thuộc tính này:
 - + Border-box (mặc định) - ảnh nền được vẽ từ viền bên ngoài của đường viền
 - + Padding-box - ảnh nền được vẽ từ bên ngoài viền của padding
 - + Content-box - ảnh nền được vẽ với trong content box

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #example1 { border: 10px dotted black; padding: 35px; background: yellow; } </style> </head> <body> <h1>The background-clip Property</h1> <p>No background-clip (border-box is default):</p> <div id="example1"> <h2>Lorem Ipsum Dolor</h2> </div> </body> </html></pre>	<p>The background-clip Property</p> <p>No background-clip (border-box is default):</p> 

19. CSS Buttons

19.1. Basic Button Styling

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer; } </style> </head> <body> <h2>CSS Buttons</h2> <button>Default Button</button> <class="button">Link Button <button class="button">Button</button> <input type="button" class="button" value="Input Button"> </body> </html></pre>	<p>CSS Buttons</p>

19.2. Button Colors

- Sử dụng thuộc tính background-color để thay đổi màu sắc của một nút nhấn.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /*Green*/ border: none; color: white; padding: 15px 32px;</pre>	<p>Button Colors</p> <p>Change the background color of a button with the background-color property:</p>

```

text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}
.button2 {background-color: #008CBA;}
/* Blue */
.button3 {background-color: #f44336;}
/* Red */
.button4 {background-color: #e7e7e7;
color: black;} /* Gray */
.button5 {background-color: #555555;}
/* Black */
</style>
</head>
<body>
<h2>Button Colors</h2>
<p>Change the background color of a
button with the background-color
property:</p>
<button class="button">Green</button>
<button class="button
button2">Blue</button>
<button class="button
button3">Red</button>
<button class="button
button4">Gray</button>
<button class="button
button5">Black</button>
</body>
</html>

```

19.3. Button Sizes

- Sử dụng thuộc tính font-size để thay đổi kích thước của một button.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; } </pre>	<p>Button Sizes</p> <p>Change the font size of a button with the font-size property:</p>

```
text-decoration: none;
display: inline-block;
margin: 4px 2px;
cursor: pointer;
}

.button1 {font-size: 10px;}
.button2 {font-size: 12px;}
.button3 {font-size: 16px;}
.button4 {font-size: 20px;}
.button5 {font-size: 24px;}
</style>
</head>
<body>

<h2>Button Sizes</h2>

<p>Change the font size of a
button with the font-size
property:</p>

<button class="button
button1">10px</button>
<button class="button
button2">12px</button>
<button class="button
button3">16px</button>
<button class="button
button4">20px</button>
<button class="button
button5">24px</button>

</body>
</html>
```

- Sử dụng thuộc tính padding để thay đổi padding của một nút nhấn.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; text-align: center; text-decoration: none; }</pre>	<p>Button Padding</p> <p>Change the padding of a button with the padding property:</p> <p>10px 24px 12px 28px</p> <p>14px 40px 32px 16px 16px</p>

```
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}

.button1 {padding: 10px 24px;}
.button2 {padding: 12px 28px;}
.button3 {padding: 14px 40px;}
.button4 {padding: 32px 16px;}
.button5 {padding: 16px;}
```

</style>

</head>

<body>

<h2>Button Padding</h2>

<p>Change the padding of a button with the padding property:</p>

<button class="button button1">10px 24px</button>

<button class="button button2">12px 28px</button>

<button class="button button3">14px 40px</button>

<button class="button button4">32px 16px</button>

<button class="button button5">16px</button>

</body>

</html>

19.4. Rounded Buttons

- Sử dụng thuộc tính border-radius để bo tròn các góc của một nút nhấn.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 20px; border-radius: 50%; /* Add rounded corners */ }</pre>	<p>Rounded Buttons</p> <p>Add rounded corners to a button with the border-radius property:</p> 

```

text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}
.button1 {border-radius: 2px;}
.button2 {border-radius: 4px;}
.button3 {border-radius: 8px;}
.button4 {border-radius: 12px;}
.button5 {border-radius: 50%;}
</style>
</head>
<body>
<h2>Rounded Buttons</h2>
<p>Add rounded corners to a button with the border-radius property:</p>
<button class="button button1">2px</button>
<button class="button button2">4px</button>
<button class="button button3">8px</button>
<button class="button button4">12px</button>
<button class="button button5">50%</button>
</body>
</html>

```

19.5. Colored Button Borders

- Sử dụng thuộc tính border để thêm màu viền cho nút nhấn của bạn.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; } </pre>	<p>Colored Button Borders</p> <p>Use the border property to add a border to the button:</p>  <p>The result shows five rectangular buttons arranged in two rows. The top row contains three buttons with colored borders: a green button labeled "Green", a blue button labeled "Blue", and a red button labeled "Red". The bottom row contains two gray buttons with black outlines: a larger one labeled "Gray" and a smaller one labeled "Black".</p>

Thực tập Kiến trúc và Giao thức IoT

```
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}
.button1 {
background-color: white;
color: black;
border: 2px solid #04AA6D;
}
.button2 {
background-color: white;
color: black;
border: 2px solid #008CBA;
}
.button3 {
background-color: white;
color: black;
border: 2px solid #f44336;
}
.button4 {
background-color: white;
color: black;
border: 2px solid #e7e7e7;
}
.button5 {
background-color: white;
color: black;
border: 2px solid #555555;
}
</style>
</head>
<body>

<h2>Colored Button Borders</h2>
<p>Use the border property to
add a border to the button:</p>
<button class="button
button1">Green</button>
<button class="button
button2">Blue</button>
<button class="button
button3">Red</button>
<button class="button
button4">Gray</button>
<button class="button
button5">Black</button>

</body>
</html>
```

19.6. Hoverable Buttons

- Sử dụng :hover để thay đổi trạng thái của button khi di chuột vào nó.
- Gợi ý: có thể sử dụng thuộc tính transition-duration để xác định thời gian của hiệu ứng hover.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 16px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; transition-duration: 0.4s; cursor: pointer; } .button1 { background-color: white; color: black; border: 2px solid #04AA6D; } .button1:hover { background-color: #04AA6D; color: white; } .button2 { background-color: white; color: black; border: 2px solid #008CBA; } .button2:hover { background-color: #008CBA; color: white; } .button3 { background-color: white;</pre>	<p>Hoverable Buttons</p> <p>Use the :hover selector to change the style of the button when you move the mouse over it.</p> <p>Tip: Use the transition-duration property to determine the speed of the "hover" effect:</p>

```
color: black;
border: 2px solid #f44336;
}
.button3:hover {
background-color: #f44336;
color: white;
}
.button4 {
background-color: white;
color: black;
border: 2px solid #e7e7e7;
.button4:hover {background-
color: #e7e7e7;}
.button5 {
background-color: white;
color: black;
border: 2px solid #555555;
}
.button5:hover {
background-color: #555555;
color: white;
}
</style>
</head>
<body>

<h2>Hoverable Buttons</h2>

<p>Use the :hover selector to
change the style of the button
when you move the mouse over
it.</p>
<p><strong>Tip:</strong> Use the
transition-duration property to
determine the speed of the
"hover" effect:</p>

<button class="button
button1">Green</button>
<button class="button
button2">Blue</button>
<button class="button
button3">Red</button>
<button class="button
button4">Gray</button>
<button class="button
button5">Black</button>

</body>
</html>
```

19.7. Shadow Buttons

- Sử dụng thuộc tính box-shadow để thêm bóng mờ cho nút nhấn của bạn.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer; -webkit-transition- duration: 0.4s; /* Safari */ transition-duration: 0.4s; } .button1 { box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19); } .button2:hover { box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24),0 17px 50px 0 rgba(0,0,0,0.19); } </style> </head> <body> <h2>Shadow Buttons</h2> <p>Use the box-shadow property to add shadows to the button:</p> <button class="button button1">Shadow Button</button> <button class="button button2">Shadow on Hover</button> </body> </html></pre>	<p>Shadow Buttons</p> <p>Use the box-shadow property to add shadows to the button:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> Shadow Button Shadow on Hover </div>

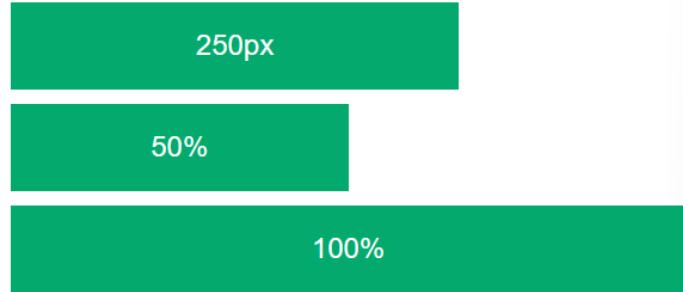
19.8. Disabled Buttons

- Sử dụng thuộc tính opacity để thêm transparency cho một nút nhán.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer; } .disabled { opacity: 0.6; cursor: not-allowed; } </style> </head> <body> <h2>Disabled Button</h2> <p>Use the opacity property to add some transparency to a button (make it look disabled):</p> <button class="button">Normal Button</button> <button class="button disabled">Disabled Button</button> </body> </html></pre>	<p>Disabled Button</p> <p>Use the opacity property to add some transparency to a button (make it look disabled):</p> <div style="display: flex; justify-content: space-around; align-items: center;"> Normal Button Disabled Button </div>

19.9. Button Width

- Mặc định, độ rộng của button được xác định bằng chiều dài nội dung của nó. Để thay đổi độ rộng này cần sử dụng thuộc tính width.

Code	Result
<pre data-bbox="166 381 714 2021"><!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer; } .button1 {width: 250px;} .button2 {width: 50%;} .button3 {width: 100%;} </style> </head> <body> <h2>Set Button Widths</h2> <p>Use the width property to change the width of the button:</p> <button class="button button1">250px</button>
 <button class="button button2">50%</button>
 <button class="button button3">100%</button> <p>Tip: Use pixels if you want to set a fixed width and use percent for responsive buttons (e.g. 50% of its parent element). Resize the browser window to see the effect.</p> </body> </html></pre>	<p data-bbox="770 381 1453 943">Set Button Widths</p> <p data-bbox="770 460 1453 494">Use the width property to change the width of the button:</p>  <p data-bbox="770 864 1453 943">Tip: Use pixels if you want to set a fixed width and use percent for responsive buttons (e.g. 50% of its parent element). Resize the browser window to see the effect.</p>

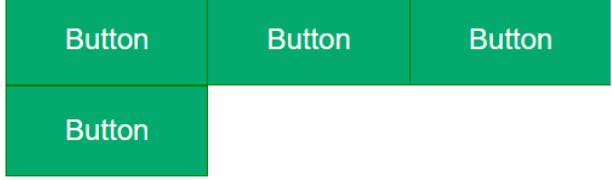
19.10. Button Groups

- Loạt bỏ đường viền và thêm float:left cho mỗi nút nhấn để tạo một nhóm các nút nhấn.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .btn-group .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; cursor: pointer; float: left; } .btn-group .button:hover { background-color: #3e8e41; } </style> </head> <body> <h2>Button Groups</h2> <p>Remove margins and float the buttons to create a button group:</p> <div class="btn-group"> <button class="button">Button</button> <button class="button">Button</button> <button class="button">Button</button> <button class="button">Button</button> </div> <p>Remember to clear floats after, or else this p element will also float next to the buttons.</p> </body> </html></pre>	<p>Button Groups</p> <p>Remove margins and float the buttons to create a button group:</p> <div style="background-color: #04AA6D; color: white; padding: 5px; text-align: center;">ButtonButtonButtonButton</div> <p>Remember to clear floats after, or else this p element will also float next to the buttons.</p>

19.11. Bordered Button Group

- Sử dụng thuộc tính border để tạo một nhóm các nút nhấn có đường viền.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> .btn-group .button { background-color: #04AA6D; /* Green */ border: 1px solid green; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; cursor: pointer; float: left; } .btn-group .button:not(:last-child) { border-right: none; /* Prevent double borders */ } .btn-group .button:hover { background-color: #3e8e41; } </style> </head> <body> <h2>Bordered Button Group</h2> <p>Add borders to create a bordered button group:</p> <div class="btn-group"> <button class="button">Button</button> <button class="button">Button</button> <button class="button">Button</button> <button class="button">Button</button> </div> </pre>	<p>Bordered Button Group</p> <p>Add borders to create a bordered button group:</p>  <p>Remember to clear floats after, or else will this p element also float next to the buttons.</p>

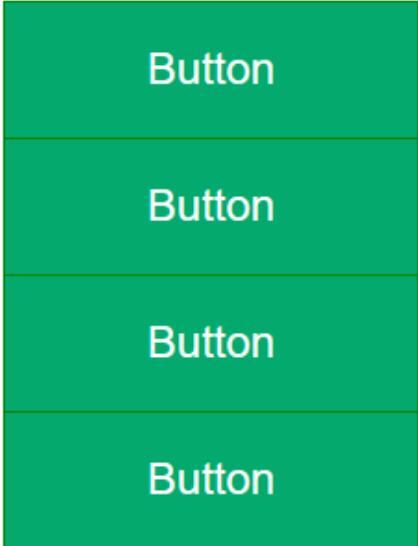
```
</div>

<p style="clear:both"><br>Remember
to clear floats after, or else will
this p element also float next to
the buttons.</p>

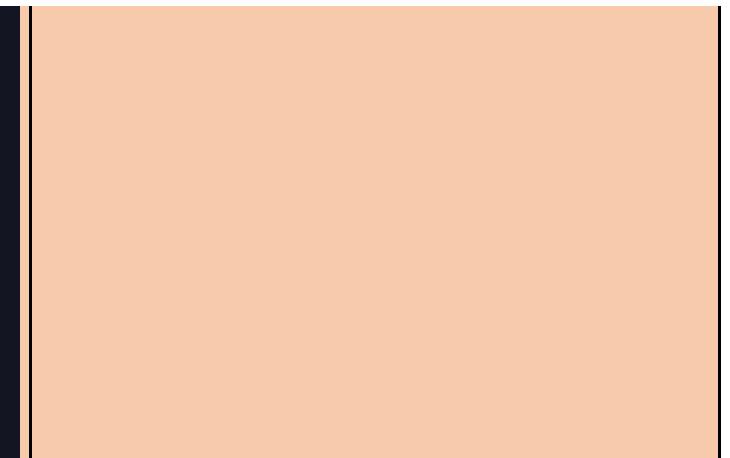
</body>
</html>
```

19.12. Vertical Button Group

- Sử dụng display:block thay cho float:left để nhóm các nút nhấn bên dưới các nút còn lại.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .btn-group .button { background-color: #04AA6D; /* Green */ border: 1px solid green; color: white; padding: 15px 32px; text-align: center; text-decoration: none; font-size: 16px; cursor: pointer; width: 150px; display: block; } .btn-group .button:not(:last-child) { border-bottom: none; /* Prevent double borders */ } .btn-group .button:hover { background-color: #3e8e41; } </style> </head> <body> <h2>Vertical Button Group</h2> <div class="btn-group"></pre>	<p>Vertical Button Group</p> 

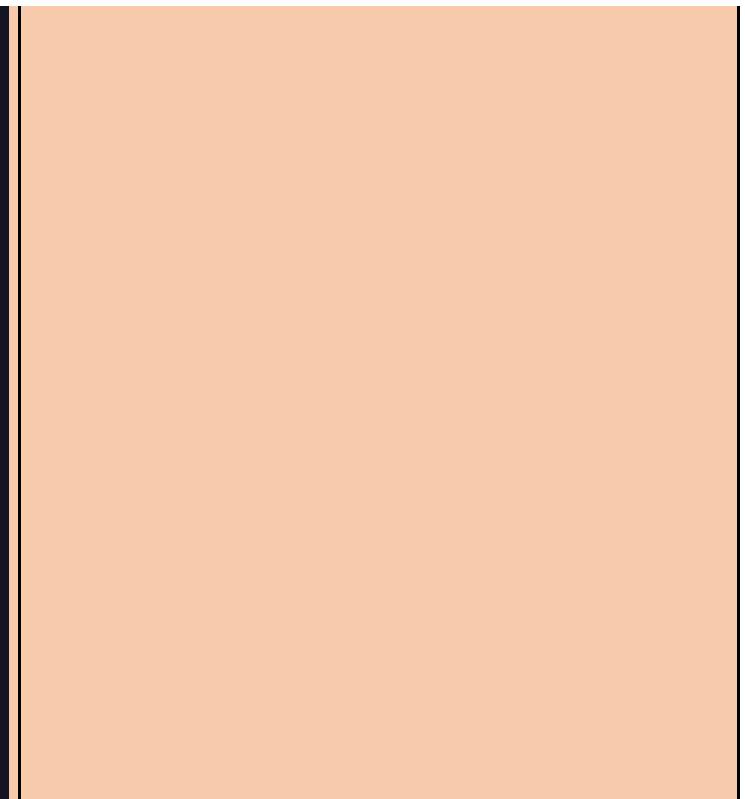
```
<button  
class="button">Button</button>  
<button  
class="button">Button</button>  
<button  
class="button">Button</button>  
<button  
class="button">Button</button>  
<button  
class="button">Button</button>  
</div>  
  
</body>  
</html>
```



19.13. Button on Image

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .container { position: relative; width: 100%; max-width: 400px; } .container img { width: 100%; height: auto; } .container .btn { position: absolute; top: 50%; left: 50%; transform: translate(-50%, - 50%); -ms-transform: translate(-50%, -50%); background-color: #f1f1f1; color: black; font-size: 16px; padding: 16px 30px; border: none; cursor: pointer; border-radius: 5px; text-align: center; }</pre>	<p>Button on Image</p> <p>Add a button on an image:</p> 

```
.container .btn:hover {  
    background-color: black;  
    color: white;  
}  
</style>  
</head>  
<body>  
  
<h2>Button on Image</h2>  
  
<p>Add a button on an image:</p>  
  
<div class="container">  
      
    <button  
        class="btn">Button</button>  
</div>  
  
</body>  
</html>
```



19.14. Animated Buttons

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .button { display: inline-block; border-radius: 4px; background-color: #f4511e; border: none; color: #FFFFFF; text-align: center; font-size: 28px; padding: 20px; width: 200px; transition: all 0.5s; cursor: pointer; margin: 5px; } .button span { cursor: pointer; display: inline-block; position: relative; transition: 0.5s;</pre>	<p>Animated Button</p> A screenshot of a web browser showing an orange button with the text "Hover »" on it. The button has a rounded rectangular shape and a slight shadow. A mouse cursor is hovering over the button, causing it to change color and appear slightly darker.

```
}

.button span:after {
  content: '\00bb';
  position: absolute;
  opacity: 0;
  top: 0;
  right: -20px;
  transition: 0.5s;
}

.button:hover span {
  padding-right: 25px;
}

.button:hover span:after {
  opacity: 1;
  right: 0;
}
</style>
</head>
<body>

<h2>Animated Button</h2>

<button class="button"
style="vertical-
align:middle"><span>Hover
</span></button>

</body>
</html>
```

20. CSS Object- position

- Thuộc tính object-position của CSS được sử dụng để chỉ định cách thức một hình ảnh, video nên được đặt ở đâu trong khu vực bao quanh nó.

20.1. The Image

- Chúng ta sử dụng thuộc tính object-fit: cover; để giữ lại tỉ lệ khung hình và kích thước nhất định.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> img { width: 200px; height: 300px; object-fit: cover; } </style> </head> <body> <h2>Using object-fit: cover</h2> </body> </html></pre>	<p>Using object-fit: cover</p>

20.2. Using the object-position Property

- Hình ảnh trên chưa được đặt ở vị trí chúng ta muốn, để thiết lập vị trí cho ảnh đó ta cần dùng thuộc tính object-position.
- Chúng ta dùng thuộc tính object-position để điều chỉnh hình ảnh vì vậy tòa công trình cũ sẽ ở tâm trang web.

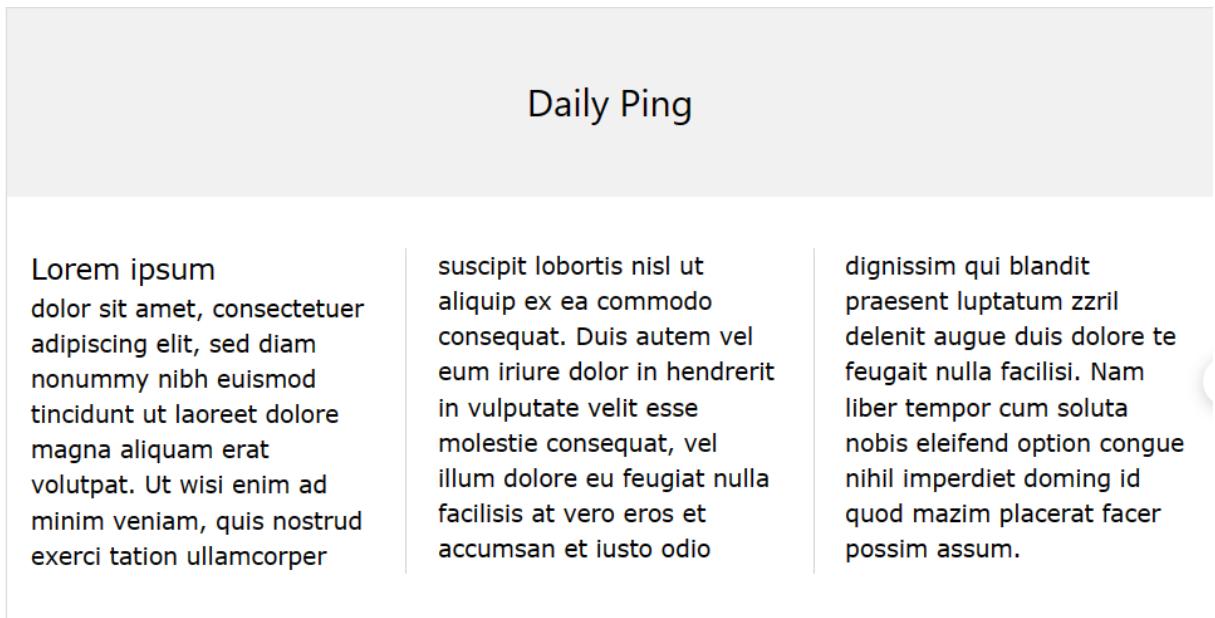
Code	Result
<pre><!DOCTYPE html> <html> <head> <style> img { width: 200px; height: 300px; object-fit: cover; object-position: 80% 100%; } </style> </head> <body> <h2>Using object-position</h2> <p>Here we will use the object- position property to position the</pre>	<p>Using object-position</p> <p>Here we will use the object-position property to position the image so that the great old building is in center:</p>

```
image so that the great old building  
is in center:</p>  
  
  
  
</body>  
</html>
```

21. CSS Multiple Columns

21.1. CSS Multi-column Layout

- Bổ cục nhiều cột cho phép dễ dàng định nghĩa cho các cột văn bản, giống như trên tạp chí.



21.2. CSS Create Multiple Columns

- Thuộc tính column-count chỉ định số cột của một phần tử sẽ được chia.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .newspaper { column-count: 3; } </style> </head> <body> <h1>Create Multiple Columns</h1> <div class="newspaper"></pre>	<p>Create Multiple Columns</p> <p>Lore ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.</p>

```

Lorem ipsum dolor sit amet,
consectetuer adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim
veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat. Duis
autem vel eum iriure dolor in
hendrerit in vulputate velit esse
molestie consequat, vel illum dolore
eu feugiat nulla facilisis at vero
eros et accumsan et iusto odio
dignissim qui blandit praesent
luptatum zzril delenit augue duis
dolore te feugait nulla facilisi. Nam
liber tempor cum soluta nobis eleifend
option congue nihil imperdiet doming
id quod mazim placerat facer possim
assum.
</div>

</body>
</html>

```

21.3. CSS Specify the Gap Between Columns

- Thuộc tính column-gap chỉ định khoảng cách giữa các cột.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> .newspaper { column-count: 3; column-gap: 40px; } </style> </head> <body> <h1>Specify the Gap Between Columns</h1> <div class="newspaper"> Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat </pre>	<p>Specify the Gap Between Columns</p> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. </p>

```

volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
</div>

</body>
</html>

```

21.4. CSS Column Rules

- Thuộc tính column-rule-style chỉ định định dạng giữa các cột.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> .newspaper { column-count: 3; column-gap: 40px; column-rule-style: solid; } </style> </head> <body> <h1>Add a Rule Between the Columns</h1> <div class="newspaper"> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut </pre>	<p>Add a Rule Between the Columns</p> <p>loboratis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.</p>

aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

</div>

</body>

</html>

- Thuộc tính column-rule-width chỉ định độ rộng của dải phân cách các cột.

eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

</div>

</body>

</html>

- Thuộc tính column-rule-color chỉ định màu của dải phân cách giữa các cột.

```
dignissim qui blandit praesent
luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
Nam liber tempor cum soluta nobis
eleifend option congue nihil
imperdiet doming id quod mazim
placerat facer possim assum.
</div>

</body>
</html>
```

- Thuộc tính column-rule là một cách viết tắt cho tất cả các thành phần tạo nên dải phân cách cho các cột.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .newspaper { column-count: 3; column-gap: 40px; column-rule: 1px solid lightblue; } </style> </head> <body> <h1>Use the column-rule Shorthand Property</h1> <div class="newspaper"> Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.</pre>	<h2>Use the column-rule Shorthand Property</h2> <p>Use the column-rule Shorthand Property</p> <p>loboris nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.</p>

```

Nam liber tempor cum soluta nobis
eleifend option congue nihil
imperdiet doming id quod mazim
placerat facer possim assum.
</div>

</body>
</html>

```

21.5. Specify How Many Columns an Element Should Span

- Thuộc tính column-span chỉ định một phần tử nên trải dài trên bao nhiêu cột.

Code	Result		
<pre> <!DOCTYPE html> <html> <head> <style> .newspaper { column-count: 3; column-gap: 40px; column-rule: 1px solid lightblue; } h2 { column-span: all; } </style> </head> <body> <div class="newspaper"> <h2>Lorem Ipsum Dolor Sit Amet</h2> Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui </pre>	Lorem Ipsum Dolor Sit Amet Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui		

```
dignissim qui blandit praesent
luptatum zzril delenit augue
duis dolore te feugait nulla
facilisi. Nam liber tempor cum
soluta nobis eleifend option
congue nihil imperdiet doming
id quod mazim placerat facer
possim assum.
</div>

</body>
</html>
```

21.6. Specify The Column Width

- Thuộc tính column-width chỉ định, tối ưu độ rộng của các cột.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .newspaper { column-width: 100px; } </style> </head> <body> <h1>Specify The Column Width</h1> <div class="newspaper"> Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis</pre>	<h2>Specify The Column Width</h2> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. </p>

```
eleifend option congue nihil  
imperdiet doming id quod mazim  
placerat facer possim assum.  
</div>  
  
</body>  
</html>
```

III. Google Firebase

1. Giới thiệu

- Google firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google. Chức năng chính của firebase là giúp người dùng lập trình ứng dụng, phần mềm trên các nền tảng web, di động bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

- Google firebase bao gồm việc:
 - + Lưu trữ dữ liệu thời gian thực.
 - + Xác thực người dùng.
 - + Firebase hosting.

2. Ưu nhược điểm

2.1. Ưu điểm:

- Xây dựng ứng dụng nhanh chóng mà không tốn thời gian, nhân lực để quản lý hệ thống và cơ sở hạ tầng phía sau. Firebase cung cấp cho bạn chức năng như phân tích, cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cố lỗi để bạn có thể dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm đem lại các trải nghiệm tốt nhất cho họ.

- Uy tín chất lượng đảm bảo từ Google: Firebase được google hỗ trợ và cung cấp trên nền tảng phần cứng với quy mô rộng khắp thế giới, được các tập đoàn lớn và các ứng dụng với triệu lượt sử dụng từ người dùng.

- Quản lý cấu hình và trải nghiệm các ứng dụng của Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.

2.2. Nhược điểm:

- Nhược điểm lớn nhất của Firebase chính là phần Realtime Database, mà đúng hơn chỉ là phần Database. Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn từ mức độ chút tới khá nhiều.

- Firebase Realtime Database có giới hạn về quy mô và chức năng so với các cơ sở dữ liệu quan hệ truyền thống.

Thực tập Kiến trúc và Giao thức IoT

- Mặc dù Firebase cung cấp kế hoạch miễn phí, nhưng sử dụng nhiều tính năng và tài nguyên có thể tạo ra chi phí đáng kể. Cần xem xét chi phí của dự án khi phát triển ứng dụng lớn.

- Rủi ro vendor lock-in: sử dụng Firebase có thể gây ra rủi ro khóa vào nền tảng của Google, khó khăn khi bạn muốn di chuyển ứng dụng sang nền tảng khác trong tương lai.

3. Code

3.1 HMTL

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" > <!-- xác định bộ ký tự được sử dụng trong tài liệu là UTF-8 -->
    <meta http-equiv="X-UA-Compatible" content="IE=edge" > <!-- chỉ định chế độ tương thích của trình duyệt. -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" >
<!-- điều chỉnh hiển thị trên các thiết bị di động. -->
    <title>Document</title>
    <!-- liên kết tới tệp CSS của Materialize CSS framework từ một URL xa. -->
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
        <link rel="stylesheet" href=".//style.css" > <!-- liên kết tới tệp CSS nằm trong cùng thư mục với tệp HTML hiện tại -->
</head>
<body>
    <main>
        <!-- khung nhiệt độ và độ ẩm -->
        <form>
            <h1 id="nhietdo"></h1>
            <h2 id="doam"></h2><br>
        </form>
        <!-- khung hình ảnh và các label -->
        <form>
            <!-- hiển thị hình ảnh -->
            
            <button type="button" id="btn1" class="btn waves-effect waves-lights">On</button>
            <button type="button" id="btn2" class="btn waves-effect waves-lights">Off</button><br><br>
            <!-- label hiển thị text hoặc các chuỗi hiển thị trên web -->
            <label for="text">This website make by TT-IOT team</label><br>
            <label for="text">Duong Trieu Men</label><br>
            <label for="text">Chau Le Thuy Tien</label><br>
            <label for="text">Cao Tien Sy</label><br>
        </form>
    </main>
<script src="function.js"></script> <!-- liên kết với functions.js -->
```

Thực tập Kiến trúc và Giao thức IoT

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio-database.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio-analytics.js"></script>
<script>
    // Your web app's Firebase configuration
    // For Firebase JS SDK v7.20.0 and later, measurementId is optional
    const firebaseConfig = {
        apiKey: "AIzaSyCQAhE0zi3i0392TidskWPT8UaZWRB2fI",
        authDomain: "tt-iot-68173.firebaseio.com",
        databaseURL: "https://tt-iot-68173-default-rtdb.firebaseio.com",
        projectId: "tt-iot-68173",
        storageBucket: "tt-iot-68173.appspot.com",
        messagingSenderId: "176870868658",
        appId: "1:176870868658:web:cde3e326f009d71d81ebab",
        measurementId: "G-25R2JY6CHM"
    };
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);
    firebase.analytics();

    //functions
    // make sure that the name in ' ' match with name of your database child
    var nhietDo = document.getElementById('nhietdo'); //lấy tham chiếu đến phần tử có id là "nhietdo" trong DOM và lưu trữ nó trong biến nhietDo
    //tạo một tham chiếu đến mục "Nhiệt độ" trong cơ sở dữ liệu Firebase và lưu trữ nó trong biến dbRef.
    var dbRefnhietDo = firebase.database().ref().child('Nhiệt độ');
    var doAm = document.getElementById('doam');
    var dbRefdoAm = firebase.database().ref().child('Độ ẩm');
    // lắng nghe sự thay đổi dữ liệu trong tham chiếu dbRef. Khi dữ liệu thay đổi, hàm xử lý không tên được kích hoạt và nội dung của phần tử nhietDo được cập nhật thành giá trị mới (snap.val()),
    dbRefnhietDo.on('value', snap => nhietDo.innerText = snap.val());
    dbRefdoAm.on('value', snap => doAm.innerText = snap.val());
</script>

</body>
</html>
```

3.2 CSS

```
/* thiết kế thông số và định dạng trang web */
* {
    padding: 0;
```

Thực tập Kiến trúc và Giao thức IoT

```
    margin: 0;
}
main {
    width: 100vw;
    height: 100vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
form {
    width: 400px;
    border: solid 1px #E8E8E8;
    padding: 5rem;
}
button {
    margin-top: 2rem;
}
.fan_off{
    width: 200px;
    height: 200px;
    margin: auto;
}
.fan_on{
    width: 200px;
    height: 200px;
    margin: auto;
}
```

3.3 Javascript

```
// khai tao cac thong so
// khai tao cac thong so
//lấy tham chiếu đến phần tử có id là "btn1" trong DOM (Document Object
Model) và lưu trữ nó trong biến btn1.
let btn1 = document.querySelector('#btn1');
let img = document.querySelector('img');
let btn2 = document.querySelector('#btn2');

// functions nut bam
// Khi nút này được nhấp, hàm xử lý không tên (arrow function) được kích
hoạt.
btn1.addEventListener('click', ()=>{
    img.src = 'img/fan_running.png'; //thay đổi nguồn hình ảnh của phần tử
<img> thành 'img/fan_running.png'.
})
//
btn2.addEventListener('click', ()=>{
    img.src = 'img/fan_off.png';
})
```

Thực tập Kiến trúc và Giao thức IoT

THÀNH VIÊN NHÓM



GVHD: Trương Quang Phúc