



#7

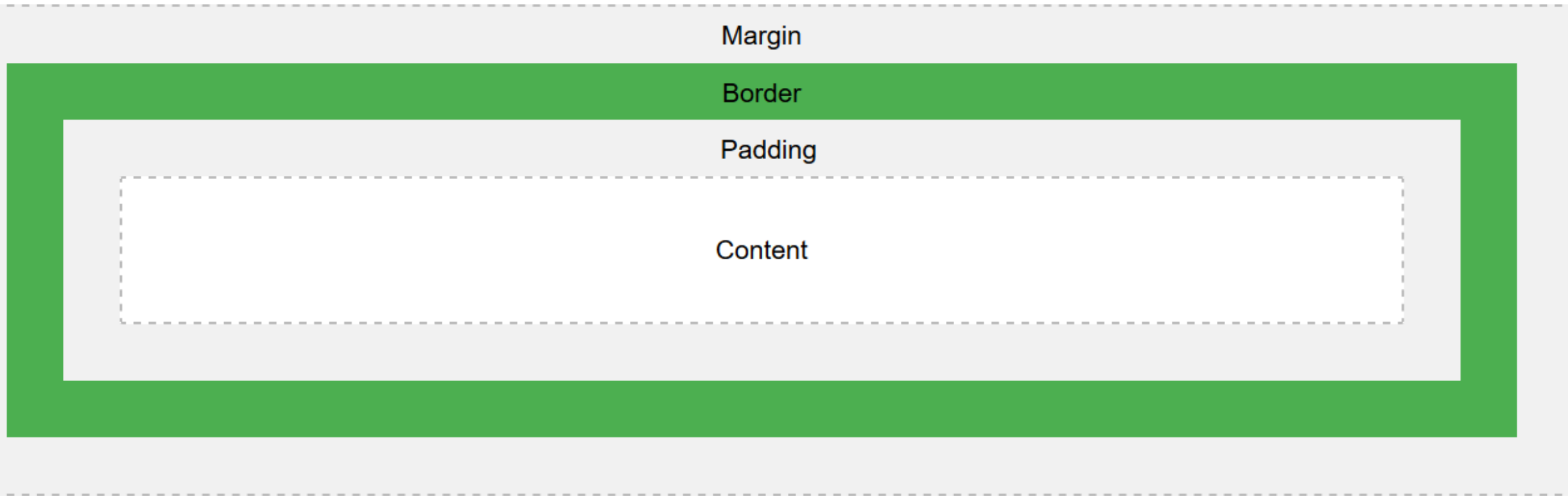
CSS:

Cascading Style Sheets

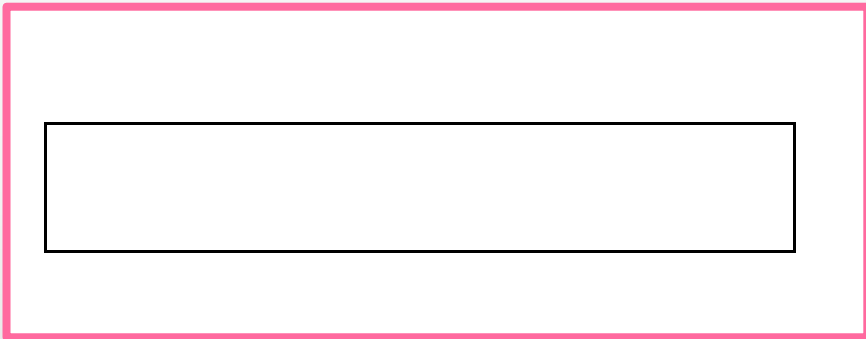
# CSS

- Describe cómo será mostrado todo el HTML
- Puede aplicarse a varios documentos **.html**
- Estilos externos se guardan como **.css**

# CSS – Box Model



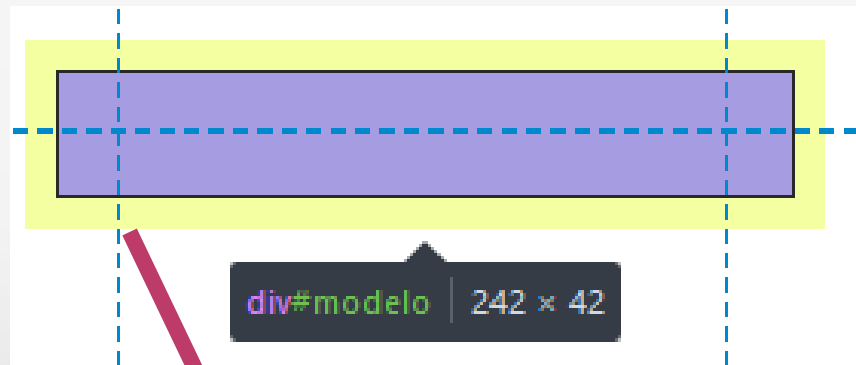
# CSS – Box Model – ¿Ancho?



```
<style>
  #modelo {
    width: 200px;
    padding: 20px;
    margin: 10px;
    border: 1px solid black;
  }
</style>

<div id="modelo"></div>
```

# CSS – Box Model – ¿Ancho?



**No** cuenta el padding

```
#modelo {  
  width: 200px;  
  padding: 20px;  
  margin: 10px;  
  border: 1px solid black;
```

200px

20px \* 2

10px \* 2

1px \* 2

# CSS – Sintaxis

- **Reglas** ← (*Estamos aquí*)
- Agrupaciones
- Herencia
- Comentarios

# CSS – Sintaxis – Reglas



- Selector
  - Qué elemento de HTML vamos a modificar
  - Puede ser por etiqueta, clase o id
- Declaración (se separan por ; (punto y coma))
  - Propiedad
  - Valor

# CSS – Sintaxis – Reglas (Selector)



*Combinen  
a gusto*

- `code { /* ... */ }`
  - Aplica solo a las etiquetas `<code>`
- `code.html { /* ... */ }`
  - Aplica sólo a las etiquetas `<code>` con `class="html"`
- `code#php { /* ... */ }`
  - Aplica sólo a las etiquetas `<code>` con `id="php"`



# CSS – Sintaxis – Reglas (Declaración)

- Propiedad

- Más de 220 propiedades

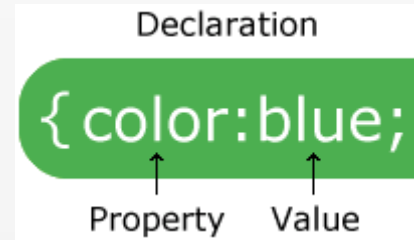
- No todas funcionan de la misma forma (por ejemplo, font-family)

- Siempre terminan con : (dos puntos)

- Valor

- Acorde a la propiedad

- Siempre cierra con ; (punto y coma)



# CSS – Sintaxis

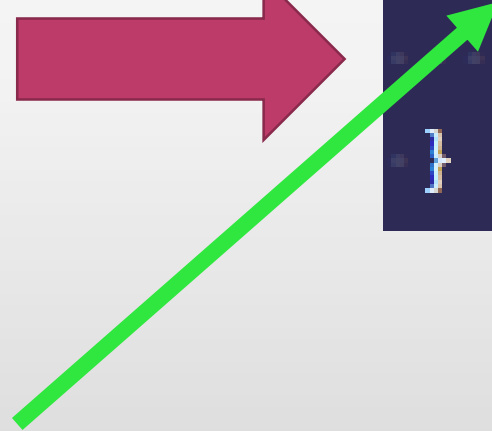
- Reglas
- **Agrupaciones** ← (*Estamos aquí*)
- Herencia
- Comentarios

# CSS – Sintaxis – Agrupaciones

```
p {  
  color: blue;  
}  
h1 {  
  color: blue;  
}
```



```
h1, p {  
  color: blue;  
}
```



Se agrupa usando , (coma)

# CSS – Sintaxis – Agrupaciones

- Respetar “jerarquía” de importancia (legibilidad)
  - $h1 > h2 > p$
- Apilar elementos repetidos de arriba hacia abajo (legibilidad)

```
h1,  
h2,  
h3,  
h4,  
h5,  
h6 {  
  color: blue;  
}
```

# CSS – Sintaxis

- Reglas
- Agrupaciones
- **Herencia** ← (*Estamos aquí*)
- Comentarios

# CSS – Sintaxis – Herencia

```
<style>
  section {
    color: red;
  }
</style>
</head>
<body>
  <p>Texto fuera de un SECTION</p>
  <section>
    <p>Texto dentro de un SECTION</p>
  </section>
</body>
```

Texto fuera de un SECTION

Texto dentro de un SECTION

Listeners >>

:hov .cls +

```
p {
  display: block;
  margin-block-start: 1em;
  margin-block-end: 1em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
}
Inherited from section
section {
  color: red;
}
index.html:11
```

# CSS – Sintaxis – Herencia

- Las etiquetas (**parent**) *que agrupan otras etiquetas* (**child**) heredan las propiedades de su selector externo
  - A menos que se especifique lo contrario...
- Es por esto que se llama “**en cascada**”
- Excepciones...

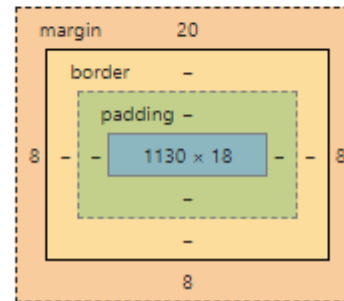
# CSS – Sintaxis – Herencia

```
body {  
  margin-top: 20px;  
}
```

```
<body>  
<p>YOU HAVE NO POWER HERE</p>  
</body>
```

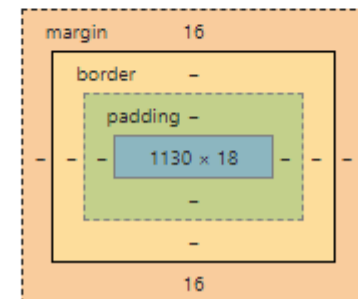
**<body>**

Styles	Computed	Event Listeners	>>
Filter	:hov .cls +		
element.style {			
body {	index.html:11		
margin-top: 20px;			
}			
body {	user agent stylesheet		
display: block;			
margin: 8px;			
}			



**<p>**

Styles	Computed	Event Listeners	>>
Filter	:hov .cls +		
element.style {			
p {	user agent stylesheet		
display: block;			
margin-block-start: 1em;			
margin-block-end: 1em;			
margin-inline-start: 0px;			
margin-inline-end: 0px;			
}			





# CSS – Sintaxis – Herencia

- Hoja de estilo externa (.css)
- Hoja de estilo interna (<style>)
- Estilos “inline” (<body **style=“color:red;”>**)

# CSS – Sintaxis – Herencia

1. Estilos del browser
2. Estilos internos y externos
3. Estilos “inline”

# CSS – Sintaxis

- Reglas
- Agrupaciones
- Herencia
- **Comentarios** ← (*Estamos aquí*)

# CSS – Sintaxis – Comentarios

```
<style>
... /* Estilos bonitos */
section {
```

```
/*
· Estilos muy bonitos
· Y muy sexys
*/
```

```
/*
· Esto queda mas bonito todavia
· Porque tiene botoncitos a la izq
*/
```

## Consejo ✨

- Nombre de clases = Acorde a la “función”
  - `<button class=“primario”>Enviar</button>`
  - `<button class=“secundario”>Cerrar</button>`
- Nombre de ID = Enfocado en elementos “especiales”
  - `<button class=“primario” id=“flotante”>Enviar Consulta</button>`

# Consejo ✨

- Por defecto Chrome muestra los diferentes CSS “heredados” de los exploradores / del SO (Sistema Operativo)
- En **Firefox**, hay que activarlo
  - “inspeccionar element”
  - **F1**

Inspector

☒ Show Browser Styles

### Default Developer Tools

- ☒ Style Editor
- ☐ Shader Editor
- ☐ Canvas
- ☒ Performance
- ☒ Memory
- ☒ Network
- ☒ Storage
- ☐ Web Audio

### Themes

☐ Dark ☒ Light

#### Inspector

- ☒ Show Browser Styles
- ☒ Truncate DOM attributes

Default color unit As Authored

### Web Console

- ☐ Enable timestamps

### Editor Preferences

- ☒ Detect indentation
- ☒ Autoclose brackets
- ☒ Indent using spaces
- Tab size 2
- Keybindings Default

### Advanced settings

- ☒ Enable Source Maps
- ☐ Show Gecko platform data



Ahora, es tu turno

# <link>

- Apunta a recursos externos (hojas de estilo o *stylesheets*)
- **No se cierra**



# Atributos de <link>

- rel
  - Especifica la relación entre el “recurso” y el documento (stylesheet, author, license, icon...)
- type
  - El tipo de recurso (por lo general basta con “text/css”)
- href
  - Dirección del recurso (local o remoto)

# A editar

- Propiedades:
  - font-family
  - font-weight
  - width
  - margin
  - background-color
  - ...

## Formulario AFIP 420

Este formulario te va a robar la vida

- [Seccion 1](#)
- [Seccion 2](#)
- [Seccion 3](#)

### Datos personales

Nombre y Apellido

Fecha de Nacimiento

Edad

Tipo de sangre

☐ O ☐ A ☐ B ☐ AB ☐

RH de Sangre

Email

Número de teléfono

dni

Tarjeta de Credito

### Datos de su mascota

Nombre