

UNIT 6


HTML + CSS

Markup Languages and
Information Management Systems



INDEX (part II)



- 12. Forms (I)
 - 13. Forms (II)
 - 14. CSS Generalities
 - 15. Typography
 - 16. Units of Measurement
 - 17. Box Model
- 

12. FORMS (I)



UNIT 6 HTML + CSS, PART II



Forms (I)

- A form is a document intended to **collect data** in a structured way.
- It is a means of **communication with the user**.
- Digital forms are more **versatile** than traditional ones.

**FORMULARIO PARA OFENDIDOS
EN REDES SOCIALES**

FECHA DE LA OFENSA.....HORA.....RED SOCIAL.....

¿QUE LO OFENDIO ?

UN MEME ☐ UNA PUBLICACIÓN ☐ UN ENLACE ☐ UNA FOTO ☐

UN VIDEO ☐ ALGUIEN QUE NO PIENSA COMO USTED ☐

ALGUNA OTRA:

¿LLORÓ?

MUCHO ☐ POQUITO ☐ A MOCO SUELTO ☐ COMO UNA MAGDALENA ☐

¿LO VA A SUPERAR?

SI ☐ NO ☐ NO LO SE ☐ TAL VEZ CON TERAPIA ☐ JAMAS ☐

MOTIVOS POR LOS QUE SE OFENDIÓ

<input type="radio"/> SOY UN IDIOTA.	<input type="radio"/> TENGO MUCHOS COMPLEJOS.
<input type="radio"/> ESTABA ESTALKEANDO	<input type="radio"/> RELIGION
<input type="radio"/> ME GUSTA EL DRAMA.	<input type="radio"/> SOY UN MOJIGATO.
<input type="radio"/> NECESITO ATENCIÓN.	<input type="radio"/> TODOS ESTÁN EN MI CONTRA.
<input type="radio"/> SOY DE CRISTAL.	<input type="radio"/> PREFERENCIA SEXUAL
<input type="radio"/> ME TOMO TODO A TITULO PERSONAL.	<input type="radio"/> LA TIERRA ES PLANA.
<input type="radio"/> NO SE ME OCURRIO A MI.	<input type="radio"/> SOY DE UNA MINORÍA QUE NADIE TOMA EN SERIO.
<input type="radio"/> A MIS SELFIES NADIE LES DA LIKE.	<input type="radio"/> FOTO DE PERFIL FEA

¿COMO PODEMOS MEJORAR LAS REDES PARA SU CONFORT?

☐ ACEPTANDO SOLO GENTE NORMAL COMO USTED

☐ HACIENDO PRUEBAS DE ADN

☐ DEJAR QUE USTED CENSURE HASTA A SU MADRE

☐ DARLE TODAS LAS CONTRASEÑAS Y QUE LA SUYA SEA PRIVADA

☐ LAS NORMAS DE LA COMUNIDAD LAS DICTE USTED Y SOLO USTED DEPENDIENDO DE SU ESTADO DE ANIMO.

FORMS (I)



Controls

- Text box
- Password box
- Text area
- Check boxes
- Radio buttons
- Pick list
- Submit button
- Reset button
- Image button
- Regular button
- Attach file button
- Hidden field
- Others (HTML5)

Forms (I)



- Users will enter data, pick options... by using the form **controls**.
- Two **ways of handling the collected information**:
 - Sending by **email**.
 - Processing. To do this:
 - It must be stored in a DB, XML file, JSON file...
 - A **CGI** (*Command Gateway Interface*) is required to send it to the DB.
 - The CGI program must also be present on the web server.
 - A CGI can be coded using PERL, PHP, ASP, C++...

Forms (I)

Structure of an HTML form:

- `<form>` opening tag.
 - Form body (text and controls).
 - Submit button.
 - Optionally, reset button.
- `</form>` closing tag.
- All data collected in the controls within the `<form>` element will be sent/processed after the submit button is pressed.
 - Controls could also be outside a `<form>` element, but they would be *useless*.



Forms (I)

Not to be mistaken for "user-agent"

ATTRIBUTES

Important `<form>` attributes (I):

- **action**: specifies the name of the **agent** that will receive the data. Values:
 - If sent by email:
`mailto:someone@example.com`
 - If processed by a CGI script:
URI to the script file
- **enctype** (*post method only, see next slide*): specifies how the information is encoded. Values:
 - `application/x-www-form-urlencoded`: default value; characters are encoded.
 - `multipart/form-data`: required value if the form submission has files attached.
 - `text/plain`: characters are not encoded (discouraged value).

Forms (I)

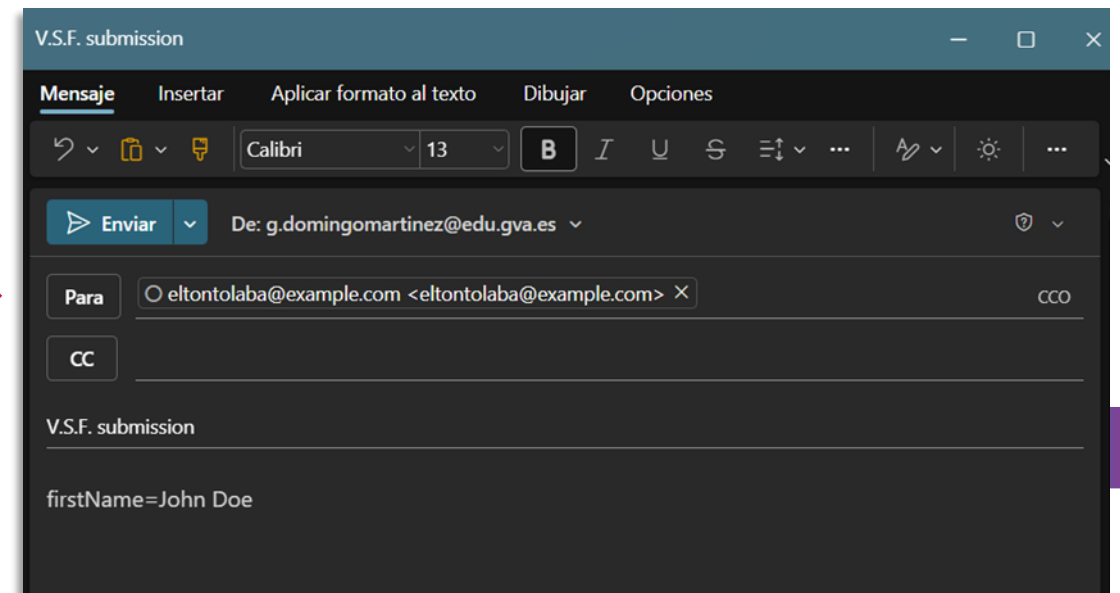
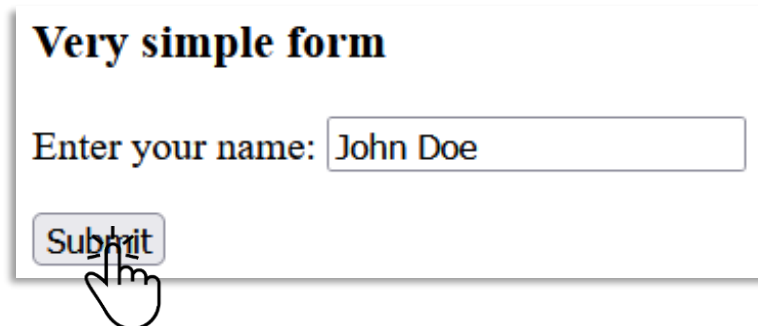
Important `<form>` attributes (II):

- **method**: specifies the HTTP method used to send the form data:
 - **get**: the data is added **as a tail to the URI** assigned to **action**.
`...?name1=value1&name2=value2&...`
 - The CGI script takes the data from the URI.
 - Caution! Security and length.
 - Does not allow sending attached files.
 - Examples: Google, YouTube.
 - **post**: the data is sent via HTTP POST transaction.
 - Not visible.
 - The CGI script takes the data directly from the form.
 - Attachments are allowed.

Forms (I)

```
<h3>Very simple form</h3>
<form method="post" enctype="text/plain"
action="mailto:eltontolaba@example.com?subject=V.S.F.%20submission">
  <p>Enter your name:
  <input type="text" name="firstName" value="John Doe">
  <p><input type="submit" value="Submit">
</form>
```

HTML code



Forms (I)

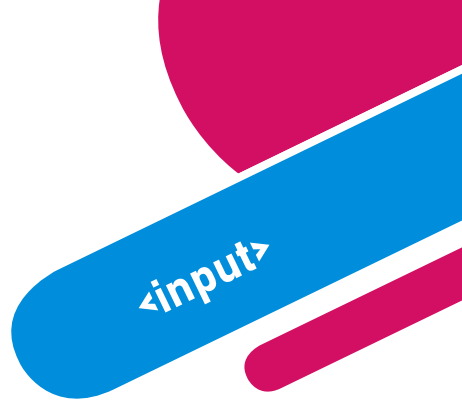
- In slide #5 you can find a list of **controls** likely to appear in a form.
- Only **three elements** are needed to add all of them:
 - `<input>`
 - `<textarea>`
 - `<select>`

FORMS

Controls

- Text box
- Password box
- Text area
- Check boxes
- Radio buttons
- Pick list
- Submit button
- Reset button
- Image button
- Regular button
- Attach file button
- Hidden field
- Others (HTML5)

Forms (I)



- General `<input>` syntax:

```
<input type="_type" name="_name" value="_value"  
      size="_size" maxlength="_maxlength" checked>
```

- Does not have a closing tag.
- Its most important attribute is `type`:
 - It determines which control is inserted into the form:
 - text
 - password
 - checkbox
 - radio
 - submit
 - reset
 - file
 - hidden
 - image
 - button
 - ...

Forms (I)

<input>

Values that the **type** attribute can take (I):

- **text**: **visible-text box**.
 - type default value.
- **password**: **obscured-text box**.
- **checkbox**: allows you to **select/deselect single values**.
- **submit**: button that **causes** the collected **info to be sent**.
- **reset**: button that **clears** the form.
 - All controls take default values.

visible text

••••••••



Submit Query

Reset

Forms (I)

Values that the `type` attribute can take (II):

- `button`: **generic button** to which actions can be associated through programming.
- `hidden`: its value is **not displayed, but submitted**.
 - Useful for sending data without the user noticing.
- `radio`: allows you to **choose one single value from many**.
- `file`: **attaches a file** that will be submitted.
- `image`: graphical **submit** button.

`<input>`

Do something



Browse... No file selected.



Forms (I)

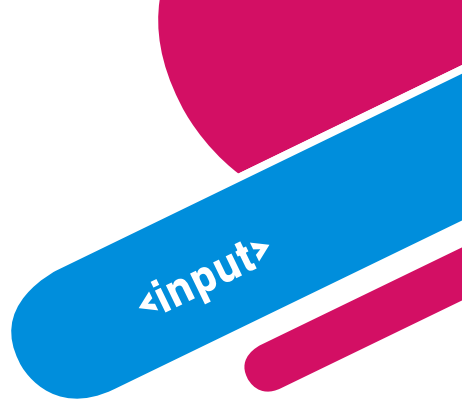
<input>

Other attributes of <input> (I):

When submitted, a key-value pair will be sent for every such controls, like name=value (see [slide #10](#))

- **name**: the name to which the value entered in that control will be associated.
 - Available for all controls, except for `submit`, `reset` and `image`.
 - Everyone else **must have it and be unique** (*special case: radio*). ← see [slide #19](#)
- **value**: its meaning depends on `type`.
 - Text boxes and `hidden`: default value.
 - `checkbox` and `radio`: value taken when submitted.
 - `submit`, `reset` and generic `button`: button text.
- **size**: length (number of characters) in text boxes.
 - Default is 20.
- **maxlength**: maximum number of characters in text boxes.
 - Default is 2¹⁹.
- **minlength**: minimum number of characters in text boxes.

Forms (I)



Other attributes of `<input>` (II):

- **placeholder**: used to give a **clue** or a description about what info is expected in the control
 - In text boxes.
 - Disappears when the user writes anything.
- **checked**: default check in radio and checkbox.
- **required**: establish the obligation to fulfill the control.
- **disabled**: disables the control.
 - Will be unusable and un-clickable.
 - Won't be submitted.
- **readonly**: its value cannot be changed.
 - Can get focus and be highlighted.
 - Will be submitted.

Special attributes that work with no value assigned (as is)

Forms (I)

<input>

```
<form action="" method="post" enctype="application/x-www-form-urlencoded"
name="myform">
  <p> User:
  <input type="text" maxlength="10" size="10" name="username">
  <p> Password:
  <input type="password" maxlength="10" size="10" name="userpass">
</form>
```

HTML code

User:

Password:

Forms (I)

<input>

```
<form action="" method="post" enctype="application/x-www-form-urlencoded"
name="myform">
  Check your favourite diseases:<p>
  <input type="checkbox" name="chkDia" value="Diarrhea">Diarrhea
  <input type="checkbox" name="chkGono" value="Gonorrhea">Gonorrhea
  <input type="checkbox" name="chkAmeno" value="Amenorrhea" checked>Amenorrhea
  <input type="checkbox" name="chkRhino" value="Rhinorrhea" checked>Rhinorrhea
</form>
```

HTML code

Check your favourite diseases:

☐ Diarrhea ☐ Gonorrhea ☒ Amenorrhea ☒ Rhinorrhea

Forms (I)

<input>

```
<form action="" method="post" enctype="application/x-www-form-urlencoded"
name="myform">
  What color was Santiago's white horse?<p>
  <input type="radio" name="rColor" value="white">White<br>
  <input type="radio" name="rColor" value="green">Green<br>
  <input type="radio" name="rColor" value="blue">Blue<br>
  <input type="radio" name="rColor" value="none" checked>None of the above
</form>
```

HTML code

Radio buttons are meant to allow the user to select only one option from many. All the radios in a group have the same name.

What color was Santiago's white horse?

- ☐ White
- ☐ Green
- ☐ Blue
- ☒ None of the above

Forms (I)

<input>

```
<form action="" method="post" enctype="multipart/form-data">
  <input type="file" name="attached"/><p>
    START&rarr;
    <input type="hidden" name="postId" value="5842">
    &larr;END<p>
    <input type="reset" value="Clear form">
    <input type="submit" value="Send form">
</form>
```

*Typical usage: keeping track
of the database element
to be modified
(the user does not care)*

HTML code

Browse... No file selected.

START→ ←END

Clear form Send form

Forms (I)

<input>

```
<form action="" method="post" enctype="application/x-www-form-urlencoded"
name="myform">
  <input type="image" src="images/purple_tentacle_s.png" alt="Submit Tentacle">
  <br>
  <input type="button" name="takeOnWorld" value="Take on the World!">
</form>
```

HTML code



*Alternative text
for image button
(accessibility)*

Take on the World!

Forms (I)

Training exercise

- Create this form.
- The user box must have a value (required control).
- When the submit button is pressed, it will take the user to `thankyou.html`:
 - Text "THANK YOU".
 - Redirection back to the form after 3 seconds.

The diagram illustrates a web form submission and redirection process. On the left, a browser window titled "Very simple form" contains a form with the following fields:

- User:** A text input field.
- Password:** A text input field.
- Sex:** Radio buttons for "Male", "Female" (selected), and "Other".
- Buttons:** "Clear form" and "Send".

A hand icon is shown clicking the "Send" button. A blue arrow points from the "Send" button to a second browser window on the right titled "THANK YOU". A pink arrow points from the "THANK YOU" window back to the "Very simple form" window, indicating a redirection. A small clock icon is placed on the arrow pointing back to the form, representing a 3-second delay. In the top right corner, a blue pill-shaped button contains the text `<input>`.

Forms (I)

<textarea>

- The <textarea> control displays a **multiline text box**.
- **Syntax:**

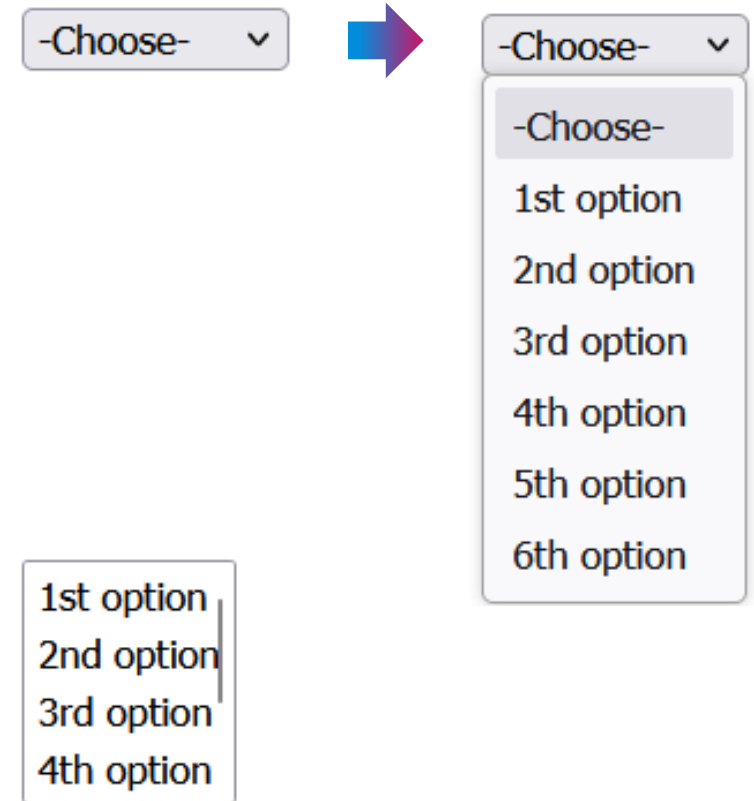
```
<textarea name="taName" cols="50" rows="4">Text</textarea>
```

A rectangular text box with a thin gray border. Inside the box, the word "Text" is written in a small, dark font at the top left. The box is empty except for this text.

- **Attributes:**
 - `cols` and `rows`: width and height of the area.
 - `placeholder`, `disabled`, `readonly`, `maxlength`, `minlength`, `name`, `required`: same as <input>.

Forms (I)

- The `<select>` control displays a vertical list of options.
- `<select>` and `</select>` delimit the **list of options**.
- `<option>` and `</option>` delimit **each option**.
- **Drop-down mode** by default.
 - Switch to **scroll mode** using the `size` attribute.
- Sometimes a *dumb option* is used as first option.



Forms (I)



<select>

- The `<select>` element **must have a name** attribute.
- An `<option>` element **may have a value** attribute.
 - If omitted, its content will be used as value when submitted.
- Other attributes of `<select>`:
 - **multiple**: allows **multiple selection** (use Ctrl or Shift for this). (As is).
 - **size**: scroll mode; its value sets the number of **options displayed**.
 - By default, **=1** (if multiple, **=4**).
 - **disabled** and **required**.
- Other attributes of `<option>`:
 - **selected**: pre-selected option(s). (As is).
 - **disabled**.

Forms (I)

```
<form action="program.php" method="post" id="fruitform">
  enctype="application/x-www-form-urlencoded"
  Pick a fruit:
  <select name="fruit" required>
    <option value="banana">Banana</option>
    <option value="persimmon" disabled>Persimmon</option>
    <option value="kiwi">Kiwi</option>
    <option value="orange" selected>Orange</option>
    <option value="peach">Peach</option>
  </select>
</form>
```

HTML code

Pick a fruit:



Pick a fruit:

- Banana
- Persimmon
- Kiwi
- Orange
- Peach

<select>

Forms (I)

```
<select name="cars" size=8>
  <optgroup label="Swedish cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="polestar">Polestar</option>
  </optgroup>
  <optgroup label="German cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
    <option value="vw">Volkswagen</option>
  </optgroup>
</select>
```

HTML code

- **Option groups** can be created within the `<select>` element.
 - Using `<optgroup>`.
 - Its **label** attribute sets the group title (un-clickable)



Swedish cars
Volvo
Saab
Polestar
German cars
Mercedes
Audi
Volkswagen

<select>

Forms (I)

IMPROVEMENTS

- Users can **sequentially focus** the interactive elements by pressing the **Tab** key.
- The **natural sequence** follows the **order of appearance** of the controls in the document.
- That sequence can be modified through attributes:
 - **autofocus**: this element will get the **focus on page load** (as is).
 - **tabindex**: can take integer values.
 - Default is 0 (some browser buttons and address box have this value too).
 - Negative value → **skipped** element when tabbing (but still focusable).
 - Non-negative value:
 - **lower value** will be focused **first**.
 - **ascending** order after that.
 - in case of coincidence, order of appearance prevails.
- **title** attribute:
 - Makes the browser display its value as a **tooltip** next to the mouse pointer.
 - User must point for one second over the control for it to display.

Forms (I)

IMPROVEMENTS

- `<fieldset>` can be used to enclose a group of controls in a box.
 - Aesthetic improvement.
 - Its title is set through a `<legend>` element.

```
<fieldset>
  <legend>Group name</legend>
  User<br>
  <input name="username" size=15 tabindex=3><br>
  Password<br>
  <input type=password name="userpass" size=12
  tabindex=2 title="Min. 8 characters">
</fieldset>
```

HTML code

Forms (I)

IMPROVEMENTS

- The `<label>` element **ties a control in with its accompanying text**.
 - The control will also activate when that text is clicked.
 - Accessibility improvement: increased clickable area; screen readers.
 - **Implicit binding**: both control and text are placed inside the `<label>` element.

```
<label>  
  <input type=checkbox name=chkSports value="Sports">Sports  
</label>
```

- **Explicit binding**: the text is placed inside the `<label>` element; the control is referred through the **for** attribute.

- Referred control **must have an id**.

```
<input type="radio" name="rSex" value="male" checked id="id_male">  
<label for="id_male">Male</label>
```

13. FORMS (II)

UNIT 6 HTML + CSS, PART II



Forms (II)

- Given the importance of data collection, when HTML5 was released, it came with many more controls for forms.
 - The number of controls is **doubled**.
 - new `<input>` types.
 - `<input>` has **new attributes** available.
- The new controls were designed with the intention that data collection would be:
 - **simpler** → benefit for the user.
 - more **precise** → benefit for the data recipient.



Forms (II)



New controls

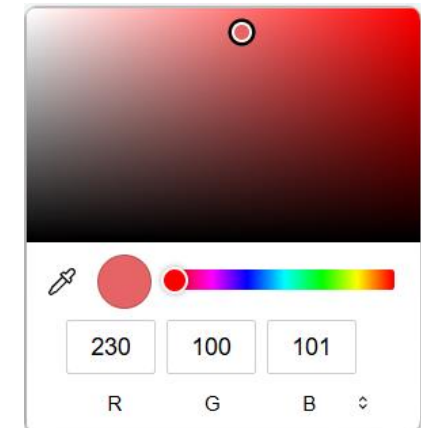
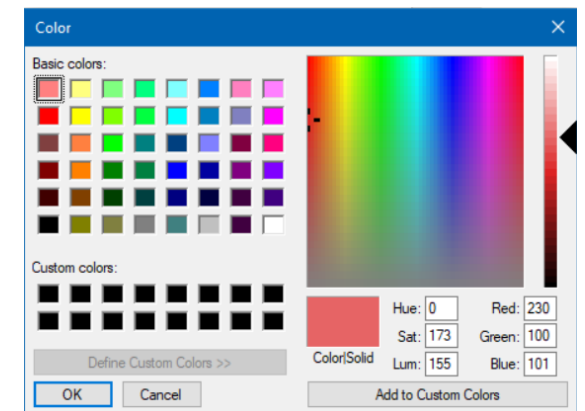
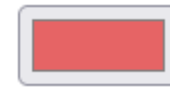
- color
- date
- datetime-local
- email
- month
- number
- range
- tel
- time
- url
- week

Forms (II)

Color picker

```
<input type="color" name="cName" value="#e66464">
```

- `value` is optional.
 - If given, format should be 7-character RGB format.
 - Otherwise (omitted or erroneous), displayed black.
- The appearance of the picker is browser-dependent.



<input>

Forms (II)

Date picker

```
<input type="date" name="birthDate">
```

- `value` attribute is **optional**.
 - If given, format should be "yyyy-mm-dd"

Birth date:

01 / 01 / 2025 📅

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	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	29	30	31	1
2	3	4	5	6	7	8

Clear

<input>

Forms (II)

Date + time picker

```
<input type="datetime-local" name="timestamp"
value="2004-09-22T16:16">
```

- `value` attribute is **optional**.
 - If given, format should be "yyyy-MM-ddThh:mm:ss"
 - If seconds are omitted, they will be zero.

Date and time of the accident:

09 / 22 / 2004 , 04 : 16 PM 📅

The image shows a date and time picker interface. At the top, there is a text input field with the value "09 / 22 / 2004 , 04 : 16 PM" and a calendar icon. Below this is a calendar for September 2004. The calendar has a grid of days from 1 to 30. The 22nd is highlighted with a blue border. At the bottom of the calendar is a "Clear" button.

<input>

Forms (II)

Month picker

```
<input type="month" name="vMonth">
```

- `value` attribute is **optional**.
 - If given, format should be "yyyy-mm"
- **Not supported by all browsers.**
 - Not supporters will display a text box instead.



The image shows the month picker UI in the Edge browser. The input field displays "enero de 2025". A calendar icon is visible on the right side of the input field, which is circled with a red dashed line. Below the input field, a dropdown menu is open, showing the year "2025" at the top. The menu contains a grid of month abbreviations: "ene.", "feb.", "mar.", "abr.", "may.", "jun.", "jul.", "ago.", "sep.", "oct.", "nov.", and "dic.". The "ene." option is highlighted with a dark background. At the bottom of the dropdown, there are two links: "Borrar" and "Este mes".



The image shows the text box UI in the Firefox browser. The input field displays the text "2025-01".

<input>

Forms (II)

Week picker

```
<input type="week" name="semana" value="2025-W01">
```

- `value` attribute is optional.
 - If given, format should be "yyyy-Www"
- Not supported by all browsers.
 - Not supporters will display a text box instead.



Semana 01, 2025

diciembre de 2024

Semana	L	M	X	J	V	S	D
48	25	26	27	28	29	30	1
49	2	3	4	5	6	7	8
50	9	10	11	12	13	14	15
51	16	17	18	19	20	21	22
52	23	24	25	26	27	28	29

1 30 31 1 2 3 4 5

Borrar Esta semana



2025-W01

<input>

Forms (II)

Time picker

```
<input type="time" name="vAppntmt" value="11:05:37">
```

- `value` attribute is optional.
 - If given, format should be "hh:mm:ss"
 - Seconds can be omitted; would be zero.



11:05:37 ⌚

11	05	37
12	06	38
13	07	39
14	08	40
15	09	41
16	10	42
17	11	43



11 : 05 : 37 AM

<input>

Forms (II)

Range slider

```
<input type="range" name="satis" min=1 max=5 step=1>
```

- Useful when:
 - A **restricted numerical value** needs to be collected.
 - The **exact value is not important**.
- **value** attribute is **optional**.
 - If omitted, it will be halfway the limits.
- **Restrictions are optional**; default values:
 - `min = 0`
 - `max = 100`
 - `step = 1`

Degree of satisfaction:



Forms (II)

Number input

```
<input type="number" name="score" min="0" max="10" step="0.01">
```

- Text box variant, **specific for numbers**.
- Optional attributes: `value`, `min`, `max`, `step`, `placeholder`, `readonly`.
 - If `step` is omitted → integer values.

Final grade:

Forms (II)

Telephone input

```
<input type="tel" id="phone" name="phone" pattern="[6-9][0-9]{8}">
```

- Text box variant, **specific for telephone numbers**.
- Optional attributes: **value**, **minlength**, **maxlength**, **size**, **placeholder**, **readonly**, **pattern**.
 - **pattern**: regular expression (RegEx) to check validity.

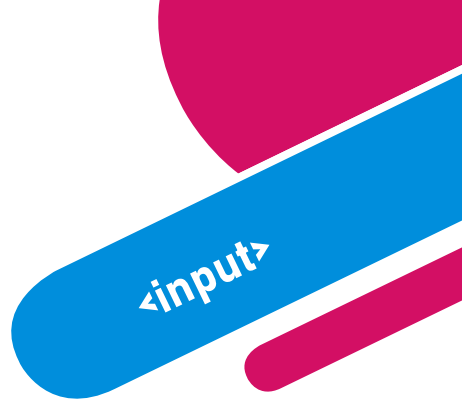
Input your phone number: 61234567

Please match the requested format.

Browsers hint the validity of the input. This number does not match the pattern.

<input>

Forms (II)



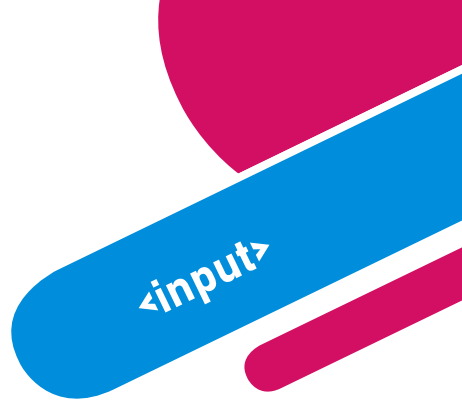
Email input

```
<input type=email name=vEmail size=64 maxlength=64 required  
  placeholder="Only @alu.edu.gva.es"  
  pattern=".+@alu\.edu\.gva\.es"  
  title="Only GVA student addresses are valid">
```

- Text box variant, **specific for email addresses**.
- Entered **value is validated** before sending.
 - Browsers have a default pattern for checking.
 - A custom **pattern** can be specified instead.
- Optional attributes: **value**, **minlength**, **maxlength**, **size**, **placeholder**, **readonly**, **pattern**, **multiple**.
 - When **multiple** is used, addresses should be entered **comma-separated**.

Input your email address:

Forms (II)



URL input

```
<input type="url" name="blog" size=30>
```

- Text box variant, **specific for URLs**.
- Entered **value is validated** before sending.
 - Browsers have a default pattern for checking.
 - A custom **pattern** can be specified instead.
- Optional attributes: **value**, **minlength**, **maxlength**, **size**, **placeholder**, **readonly**, **pattern**, **multiple**.

Personal blog:

Forms (II)

ATTRIBUTES

About the **new attributes**:

- **max** and **min**
 - Available in **number** and **date/time** related `<input>` controls.
- **pattern**
 - Available in **text**, **password**, **tel**, **url**, **email** and **search**.
 - Joint use with user aids is recommended:
 - **title** and **placeholder** attributes.
 - See table in next slide.

`<input type=search>`
*Not really intended to
collect data in a form.*



Forms (II)

ATTRIBUTES

Expresión regular	Carácter especial	Denominación	Descripción
.	Punto	Comodín	Cualquier carácter (o texto de tamaño 1)
A B	Pipe	Opciones lógicas	Opciones alternativas (o A o B)
C(A B)	Paréntesis	Agrupaciones	Agrupaciones alternativas (o CA o CB)
[0-9]	Corchetes	Rangos de caracteres	Un dígito (del 0 al 9)
[A-Z]			Una letra mayúscula de la A a la Z
[^A-Z]	^ en corchetes	Rango de exclusión	Una letra que no sea mayúscula de la A a la Z
[0-9]*	Asterisco	Cierre o clausura	Un dígito repetido 0 ó más veces (vacío incluido)
[0-9]+	Signo más	Cierre positivo	Un dígito repetido 1 ó más veces
[0-9]{3}	Llaves	Coincidencia exacta	Cifra de 3 dígitos (dígito repetido 3 veces)
[0-9]{2,4}		Coincidencia (rango)	Cifra de 2 a 4 dígitos (rep. de 2 a 4 veces)
b?	Interrogación	Carácter opcional	El carácter b puede aparecer o puede que no
\.	Barra invertida	Escape	El carácter . literalmente (no como comodín)



Forms (II)

PRACTICAL EXERCISE 6.5



14. CSS GENERALITIES



UNIT 6 HTML + CSS, PART II



CSS Generalities

CSS can be applied to elements in an HTML document in three ways:

- Inline CSS
- Internal CSS
- External CSS

Inline CSS

- CSS code is written in the HTML file, **within the specific element opening tag**.
- Highest **priority**, but most **troublesome** way.
Use only when there is no alternative and precision is required.
- **style attribute**, available for any element.

*(Obviously) selectors
are not used*

```
<p style="color: navy;  
font-family: Consolas;">  
Text here
```

HTML + CSS code

CSS Generalities

Internal CSS

- CSS code is written in the HTML file, **within the <head> block**.
 - As content of the special **<style>** element.
- 🗨️ Modifying the styles of all pages involves modifying the HTML code of each page.
- 👍 Giving a specific page a specific style that completes the generic style of the site.
- 👍 If few rules.

```
<style>
  p {color: navy;
      font-family: Consolas;
  }
</style>
```

HTML + CSS code

CSS Generalities

External CSS

- A **CSS file** containing a list of rules.
 - Plain text; .css extension.
- Linked to the HTML file through the `<link>` element, placed in the `<head>` block.
 - Multiple CSS files can be linked (using multiple `<link>` elements).

```
<link rel="stylesheet" type="text/css" href="css/estilos.css">
```

- Most used way by far.
 - Allows you to apply the same style rules to the whole site.
 - Modifying one rule affects all pages.

CSS Generalities

@-RULES

@-RULES (at-rules)

- Syntax:
`@identifier rules;`
- They provide instructions to follow or ways to behave.
- Some should be placed at the beginning of the CSS code and others can be anywhere.
- `@charset "utf-8";` should be placed at the beginning.
- Besides, you will learn `@font-face` and `@media` later on.



CSS Generalities

CASCADING

CASCADING






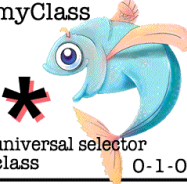

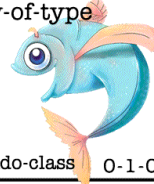
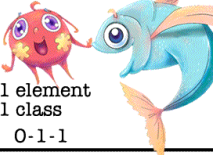
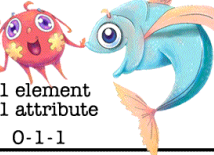
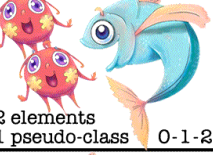
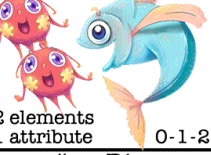
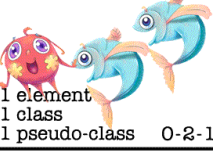
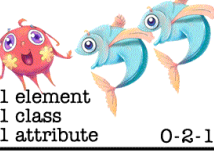
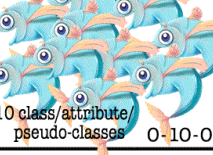
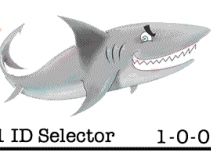
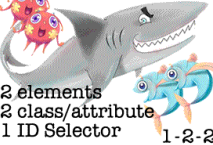
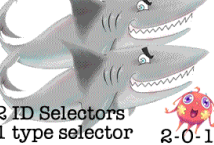


- Style rules are propagated to the children of the affected element.
- The final style for an element is the **combination** of all the properties that affect it.
- Besides, there is a **priority order** to consider:
 1. **Inline CSS** overrides both internal and external rules.
 2. A more specific selector takes precedence over a less specific one.
 - **id** selector >> **class** and **attribute** selectors >> **element** selector
 - Combinators: count single selectors **separately**; the most numerous prevails.
 3. In case of having the same specificity, rules that appear later in the code override those appearing earlier.
 4. A rule with **!important** always prevails.

*Pseudo-classes belong to the class level;
Pseudo-elements belong to the element level.
(Studied later)*

CSS Generalities

CSS SPECIFISHITY

WITH PLANKTON, FISH AND SHARKS

*  universal selector 0-0-0	div  1 element 0-0-1	li > ul  2 elements 0-0-2	body div ...ul li p a  12 elements 0-0-12
.myClass  1 class 0-1-0	*.myClass  1 universal selector 1 class 0-1-0	[type=checkbox]  1 attribute selector 0-1-0	:only-of-type  1 pseudo-class 0-1-0
li.myClass  1 element 1 class 0-1-1	li[attr]  1 element 1 attribute 0-1-1	li:nth-of-type(3n)~li  2 elements 1 pseudo-class 0-1-2	form input[type=email]  2 elements 1 attribute 0-1-2
li.class:nth-of-type(3n)  1 element 1 class 1 pseudo-class 0-2-1	input[type]:not(.class)  1 element 1 class 1 attribute 0-2-1	clnth-child(4n+1):checked  10 class/attribute/pseudo-classes 0-10-0	#myDiv  1 ID Selector 1-0-0
#myDiv li.class a[href]  2 elements 2 class/attribute 1 ID Selector 1-2-2	#divitis #myDiv a  2 ID Selectors 1 type selector 2-0-1	style=""  inline style 1-0-0-0	!important  important 1-0-0-0

X-0-0: The number of ID selectors

0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes

0-0-Z: The number of element (a.k.a. type) selectors and pseudo-elements

*, +, >, ~: Universal selector and combinators do not increase specificity

:not(x): Negation selector has no value. Argument increases specificity

ESTELLE WEYL * @ESTELLEW * WWW.STANDARDISTA.COM * 2104



15. TYPOGRAPHY



UNIT 6 HTML + CSS, PART II





わが



THANKS!

Do you have any questions?



g.domingomartinez@edu.gva.es

