

HTML

Front end
HTML structure CSS style

JS

Interactivity

Backend
Lang Python
framework Django

Database

SQLite

FORMAT →

<DOCTYPE html> → signal to use standard HTML

<html>

<head> → describe main item on page

<meta charset = "utf-8">

shows at <title> This is title! </title> → required!!
tab

</head>

<body>

<!-- Anything in is comment --> → or write text
ctrl + /

Big heading <h1> This is </h1> → h₁
on page <h2> This is </h2> → h₂ ↓ size decreases
</body> <p> </p> → para tags in body

</html>

Tags in body →

- ① `<h1></h1>` ② `<p></p>` ③ `` ④ `<i></i>` $\xrightarrow{\text{new}}$ ``
- Heading para bold \downarrow new
- ⑤ `<header></header>` → company logos, tagline, header info
- ⑥ `<nav></nav>` → links to diff website
- ⑦ `<section>` $\xrightarrow{\text{about details}}$
`<article></article>` $\xrightarrow{\text{2.2.3}}$ $\xrightarrow{\text{style}}$ $\xrightarrow{\text{HTML}}$
`</section>` $\xrightarrow{\text{content}}$
- ⑧ `<aside></aside>` → info related to main topic
- ⑨ `<footer></footer>` $\xrightarrow{\text{links}}$
- ⑩ ~~`<div>`~~
`<div class="groupone">` $\xrightarrow{\text{style}}$ $\xrightarrow{\text{CSS}}$ $\xrightarrow{\text{HTML}}$
`<h1></h1>` $\xrightarrow{\text{style}}$ $\xrightarrow{\text{HTML}}$
`<p> I want to be, styled</p>` $\xrightarrow{\text{apply effects for only 'be'}}$
`</div>`

→ list < ordered
unordered

<body>

<h1> Lists </h1>

 first

 second

 first

- second

 first

 first subpoint

{

ordered

unordered

Lists

1. First

2. Second

• First

• second

1. First

• First subpoint

* * * * *

freeze up copy

list tag absolutely

<">

nested list

→ Attributes → allow to add more info to html tags
- adding link, referencing image

① Images →
self ending tag

<image src = "wl.jpg" width = "400" height = "235"
alt = "This is the picture" >

If image in same directory

shows if image is missing
Only ↓

② linking page →

 click me

* to open in new tab
the link

 click

If picture show

↓

If doesn't

☒ This is the picture

click me → will direct
to page

③ Linking on same page

< href on same page

< section >

< a href = "#section1" > section 1

< a href = "#top" > back to top

< h1 id = "top" > links to same page </h1>

< section >

< ul >

< li > < a href = "#section1" > section1

</section>

< section id = "section1" >

< h3 > section 1 < h3 >

< p > --- </p>

</section>

→ TABLES →

< body >

< table border = "1" >

space b/w inner & outer margin

<thead >

< th > Number </th >

< th > colour </th >

table Heads

creating
row of headings
for each column

<thead >

< td > 150 </td >

< td > yellow </td >

< /td >

< /td >

< /td >

< /table >

Number	Colour
150	yellow
200	Red

FORMS →

<body>

<form>

<h1> Log in </h1>

<h2> Please input email & pass </h2>

<input type = "text" name = " " value = " " >

</form>

</body>

type of input

name of input

value of input

(assigns value to this name)

(to be prefilled)

Submit → <input type = "submit" value = "submit" >

text → <input type = "text" value = " " >

password → <input type = "password" value = " " >

email → <input type = "email" value = " " >

date → <input type = "date" value = " " >

checkbox → <input type = "checkbox" value = " " >

radio → <input type = "radio" value = " " >

file → <input type = "file" value = " " >

Action →

<form action = "url" method = "get" >

destination page

on pressing submit leads to url

label →

① <label>

Individual

Enter text:

<input type = "text" name = " " value = " " >

</label>

id → uniquely identifies the type-

②

for all

<label for = "user" > Enter Input: </label>

<input id = "user" type = " " >

→ Enter text:

* Instead of value, use placeholder.

<input type = "text" name = " " value = "Help" >

placeholder = "Help" >

→ Help
need to clear

→ Help
+ type over

* If you want to make a field mandatory to be filled before submit, use required.

<input type = "password" name = " " value = "password" required >

if submit without file → [] submit
Please fill this
popup

→ Input Methods

① Radio button ⚡

If multiple radio types
are named same, only one of those
can be selected.

```
{ <input type="radio" name="loc" value="inside">
```

```
<h2> How service? </h2>
```

Shows in list as

name
name = value

```
<select> name="stars" >
```

```
<option value="Great">3</option>
```

```
<option value="Bad">2</option>
```

```
</select> <br/>
```

Dropbox
(multiple)

```
<h2> any other feedback (rect box to write) </h2>
```

```
<textarea name="mytext" rows="8" cols="80"></textarea>
```

CSS

linking to HTML →

(In Head)

<Head>

<link rel = "stylesheet" href = "Parti-master.css" >

</head>

→ comment → /* * / → Ctrl + / ↪ repeat = 220

→ General form → selected tag { property: value; }

eg → h1 { color: blue; } changes colour of h1

↓ other way
color:rgb(8,84,1);
color:#0895a8;
color:rgba(8,84,1,0.5);

→ div, span, background → can also pass a pic by url(); instead of background: gray; changes background of body to gray

div { div { background: red; border-color: blue; border-width: 1px; border-style: dotted; } } background of only div to gray
also instead of px, can choose thin, medium, thick.

Span { background: white; border-color: black; } for inside span

* to repeat background of body (image repeats on big page)

body { background: url(---); }

background-repeat: repeat;

not repeat

if don't want to repeat

<"223.html#H109" = part of text&style = see style>

→ Selectors →

id → targets single element

classes → targets groups of elements.

text-decoration
→ line-through
text-decoration

① html <div class="name"> </div> css .name { color: blue; }

② <p id="name"> </p> css #name { color: green; }

③ <?> { color: black; } → will colour black all text except id

④ h3 + ul { border: 4px dotted purple; } can write all border together

⑤ li a { colour: red; } anchor tag

li a[href="uv.php.or"] { colour: blue; border: 5px solid orange; } specifically

→ Specificity →

<li class="n" ①

→ if

<li class="n" id="e" ②

.n {

color: blue } → ①, ②

* id has ↑ priority.

color: blue }

color: red }

bold text & coloured

red.

④ FONTS → cssfontstack.com → check availability of font

fonts.google.com → download fonts

① font-family: "Arial"; → font you know
font-family: cursive; → available in most OS

② font-size: 20px; → size of letters

Now, if we want size of 2nd para to be twice size of 1st para fonts-

body {

font-size: 20px;

}

two {

font-size: 2.0em;

}

1em = 20px

work acc
defined
to first
px

<body>

<p> - - - - - </p>

<p id="two"> - - - - </p>

</body>

③ font-style: italic; → style

④ font-weight: bold; → structure

⑤ text-align: center; → aligns center, left, right

BOX MODEL



can style left right
bottom top. is >
top = width is >

top {

border: 4px solid blue;

width: 28%; → border's length with respect to page size

text-align: center; → text is aligned at centre of border box

margin: auto; → sets border in centre of page

{ margin: 10px 40px 200px 7px;
top right bot left

bottom {

border: 4px solid red;

width: 50%;

padding: 200px;



→ 200px is the space b/w Content & border

</body>

Bootstrap

getbootstrap.com (site)

* to get bootstrap classes → getbootstrap.com / get started / copy download / CDN link.

- container class → centers everything (shifts right creates a 10px margin)

∴ various classes for various requirement

① button class (check various on [getbootstrap.com](#))

<button class=" " type="button" name="">
type of look needed (e.g. btn btn-success)
= green button

can add more class as per requirement

(i) like ~~btn~~ class="btn btn-success btn-lg"
green large

(ii) class " " disabled="disabled" ty
can't click the button

② jumbotron class → design a page with heading + para + button

<div class="container">

<div class="jumbotron" >

<h1> </h1>

<p> </p>

<p> </p>

</div> </div>

} From page link

③ form class →

① div <class = "form-group" >

provides spaces
b/w different inputs

② <input class = "form-control" >

provides design
• font highlighted
• box for email is stretched

puts email

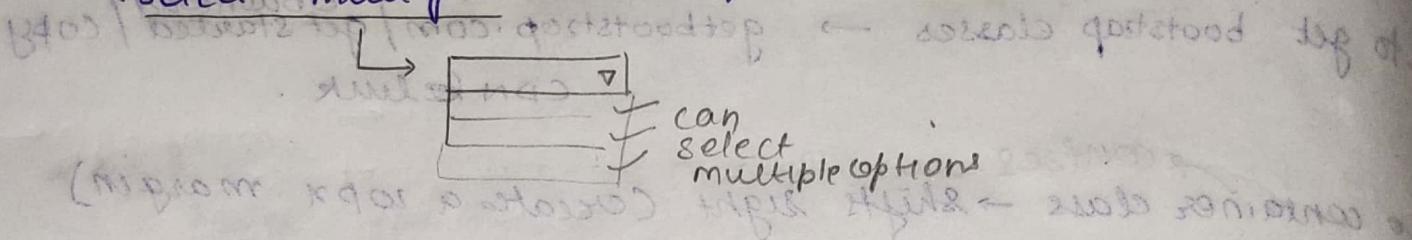
to email

③ <small id = " " class = "form-text form-muted" > </small>

email
we never share your info [para like muted text]

④ for multiple selection drop box →

<select multiple class = "form-control" ...>



⑤ text area →

* form-control also gives ability to click and stretch the area box

⑥ File upload input →

<input type="file" class = "form-control-file" id="" ...>

"choose file" NO FILE CHOSEN

click gets to open a folder

(selection - backend)

⑦ Radio button →

<legend> --- </legend>

& class = "form-check" used for radio button

* to disable any input just write <input type="radio" disabled>

Navbars → navigation bar even if scroll down, navbar is visible at top

① class = "navbar navbar-default navbar-fixed-top" in a grey tile

file = black

② class = "navbar-header"

→ main

"dropdown-menu" = sub > wh

Right click - inspect JAVASCRIPT ← use incognito tab
- console

- ① `alert("Hello World")` → string → new tab says Hello world
→ Event → OK
- ② comment → // text
- ③ All no. are treated same
- ④ strings → "text"
- ⑤ Booleans → false, true
- ⑥ undefined, null → variable created but not defined
- ⑦ to clear console → clear()
- ⑧ can do calculation → 2+2, 9/2 etc → will output
• Power → $2^{**}4 = (2^4)$
- ⑨ concatenate string → "Django" + " here" → O/P Djangohere
- ⑩ length of string → "Django".length → O/P = 6
* whitespace also counts
- ⑪ newline → \n
- ⑫ \t → tab space
- ⑬ to add quote → \" text \"
- ⑭ "Hello" [0] → O/P a
- ⑮ Variables → var varName = value;
eg var account = 100; I/P
undefined
account
100 O/P.
- eg - var greeting = "welcome back: ";
var name = "jose";
alert(greeting + name) → welcome back: jones
- ⑯ to see output in console →
`console.log("text")` → text
- ⑰ `prompt("Enter something")` → O/P
"Hello" → D/P
var age = prompt("Age")
undefined
age 50

link `js` and `html` → `<script src = " " ></script>`

→ Operators →

$3 > 2$ iff $4 \neq 4$ ~~$2 == 2$~~ "text" == "text" "2" == 2
true o/p true true true
true type check "2" == "2"
true false

$==$ → qs changes into simple datatype to check equality

$==$ → Check on datatype too

$5 != "5"$ → $5 != "5"$
false

• AND → $1 == 1 \& 2 == 2 \& 3 == 3 \rightarrow \text{true}$

• OR → $1 == 2 \mid 1 == 1 \rightarrow \text{true}$

• NOR → $!(1 == 1) \rightarrow \text{false}$ (opposite of con)

→ if else

`if () {
} else if () {
} else {
}`

→ while

a/ while condition

→ break

→ for (initialization; cond; incram)

`for (; ;) {
}`

leads out of while loop

→ functions →

`func_name (Parameter1, para2)`

`{
}`

→ In console, can call fn o/p-fn def.

console

function name → fn full

- `hello("nan")` • function `hello(name = "Frank")` → default parameter
? `console.log("Hello" + name)`, in case no para is passed
- if on console →
 - ① `hello("jose")`
O/P → Hello jose
 - ② `hello()`
O/P → Hello Frank
- function `formal(name = "Sam", title = "Sir")` →
return title + " " + name;
→ formal()
O/P Sir Sam
- global variable assigned value doesn't change if assigned in a fn.
- function-variable inside fn remains to the scope of fn.

Arrays →

`var varName = [1, 2, 3].`

* string is immutable (can't be changed by changing index)

- array name.pop() → pops last element
eg. `var lastElement = arr.pop();`
- arr.push("item") → pushes "item" in the array
- arr.length → gives length

→ 2D array → `var matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`

→ for printing array elements →
 ① `for (i=0, i<arr.length, i++)` ④ `console.log(arr)`
 - { }

→ `arr.splice(index, num)`

② `for (letter of arr)`
 - { } `console.log(letter);`

Index of Element
How many no. to delete starting from index
over elements

③ `arr.forEach(alert)` → fn name
~~arr.forEach(function(log))~~

as parameters for the fn.

④ ~~arr.indexOf(name)~~ → finds index of name