

index: faciliter l'accès à la page aux internautes en indiquant l'URL du site

<!-- ceci est un commentaire -->

/*: commentaire caché

<p> </p>: paragraphe

 </br>: passage à la ligne

(1)

<h1 (jusqu'à 6)>

: texte italique

: gras

<mark>: surligner

Liste

 : non ordonnée

 : ordonnée

éléments de la liste:

lien: ...

insertion image:

<div>: division du doc (organiser le contenu)
+ appliquer un style

****: grouper des éléments + mettre en forme

<div>: élément de bloc ≠ ****: élément de ligne

<header>: entête

<footer>: pied de la page

<nav>: liens de nav

<section>: section

<aside>: info complémentaire

<article>: Article indépendant

<figcaption> </figcaption>

Formulaire : Recueillir des données entrées

< form action = " " method = " " >

2 méthodes : Get : utilisée pour les informations non sensibles (variables dans l'URL)

: Post : informations plus sensibles et sont placées dans le corps de la requête.

Balises input :

< input type = "text" / >

type = text / password / date / email / url / number
checkbox (case à cocher) / radio (seul bouton) / submit (bouton de soumission) / reset (bouton nettoyage)

< label > : décrire ce qu'on doit rentrer

exemple 1 : champs de saisie

< label for = "pseudo" > pseudo < / label > < input type = "text" name = "pseudo" id = "pseudo" / >

exemple 2 : cases à cocher

< input type = "checkbox" name = "frite" id = "frite" / >

< label for = "frite" > Frite < / label >

exemple 3 : Boutons radio

< input type = "radio" name = "tarif" value = "jour" id = "jour" / > < label for = "jour" > jour < / label >

exemple 4 : liste déroulante

< label for = "Pays" > Pays < / label >

< select name = "Pays" >

<option Value = "tunisie"> Tunisie </option>

On peut remplacer label par **optgroup** :

Regrouper les sélections par thème.

Zone de texte : <textarea>

* **attributs de HTML** :

- Required = doit être remplie
- minlength / maxlength
- Pattern
- placeholder
- title

expressions régulières :

- : tout caractère / * : cRaine / ? : caractère / ()
grp capturant / [] : intervalle / {} : Quantificateur
- ^ négation au début de ligne / \$ fin de ligne /
- | : ou / + numérateur (au - une fois)
- {n} répète n fois / {n,m} entre n et m fois

CSS} Cascading Style Sheets

(2)

* avoir une page responsive

< head >

< meta name = "viewport" content = "device-width"

initial-scale = 1.0 >

↳ Longueur

↳ niveau de zoom

Code CSS dans la balise HTML :

< HTML style = "color: red" > texte < / HTML >

Code CSS dans < head > :

< head >

< style type = "text / css" >

div {

width = 100px

height : 50px

}

< / style >

Code CSS à part :

< head >

< link rel = "stylesheet" type = "text / css"

href = "style.css" / >

< / head >

body {

span {

div {

div {

em {

class = à mise en forme à plusieurs balises

du nom ≠ ou du même nom

id = seulement 1 nom

id : #identifiant

*class : .class

* Sélecteur enfant / Parent :

article p { } : p dans article

article > p : Premier p seulement

* NTH-CHILD :

Par exemple dans une liste :

li : nth-child (2) { } : 2ème élément

even : pair / odd : impairs (ou bien $2n / 2n+1$)

first-child / last-child



c'est quoi un Block ?

- prend la largeur disponible dans l'élément conteneur
- retour à la ligne
- espace avant / après
- contient des block / inline

exemple : body / div / h / p / article / ul / ol / form / figure / figcaption (on peut changer width / height)

c'est quoi inline ? ≠ Block mais peut contenir un inline

exemple : em / strong / span / a / button / input / label / textarea / select

padding / margin / border (-top ; -bottom ; -left ; -right)

border-radius (-top-left ect...) :

text-shadow

Arrière-plan :

background-color

background-image

Background-repeat :

repeat - x -
- y -

no repeat

Background-attachment : scroll (تجربة)
Fixed

police d'un Texte :

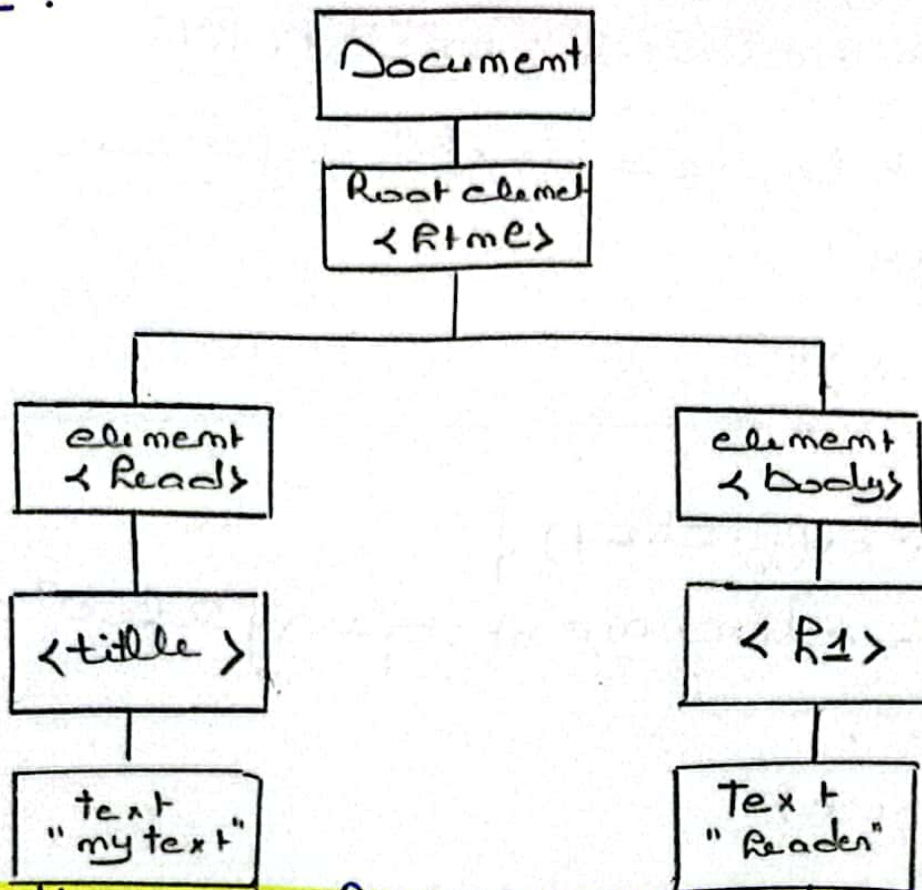
- * font-family : arial ;
- * font-style : italic ;
- * text-indent : ;
- * text-align : center / right / justify ;
- * text-decoration : underline ;
- * letter-spacing : ;

Grid

<div class="container">

- container { } (doit container display : grid;
- container div {
 - 4 colonnes
- * grid-template-columns : repeat (4, 1fr)
 - m dim ←
- * . Reader {
 - grid-area : rd;
 - }
- soit Reader (rd) / nv (nv) / contenu (ct) / footer (ft)
- grid-template-areas : (exemple)
 - " rd rd rd " (3 col Reader)
 - " nv ct ct "
- grid-gap : espace

Pour des documents HTML
exemple :



`<button onclick = "myFunction()">test</button>`
`<script>`

`function myFunction() {`

`var m = document.getElementById("t1")`

`var y = document.getElementsByClassName("a")`

`var z = document.getElementsByTagName("p")`

`}`

`</script>`

`x[0].innerHTML = "Hello world!" ;`

`C, changer le contenu`

`y[0].style.backgroundColor = "green" ;`

`z.style.color = "red";`

`document.querySelector("selector");`
`renvoie le 1er élément`


```
<script type="text/javascript" src="script.js"></script>
```

```
<form action="index.html" method="post">  
<input type="text" onfocusout="affiche()"  
id="my">  
</form>
```

HTML

JS

```
function affiche() {  
  var x = document.querySelector("#my").  
  value;  
  alert(x);  
}
```

querySelectorAll

```
<script -----></script>  
<div class="example"> - </div>  
<p class="example"> - </p>  
<button onclick="myFunction()"> - </button>
```

JS

```
function myFunction() {  
  var m, i;  
  m = document.querySelectorAll("example");  
  for (i = 0; i < m.length; i++) {  
    m[i].style.backgroundColor = "red";  
  }  
}
```


<script src = "code.js"></script>