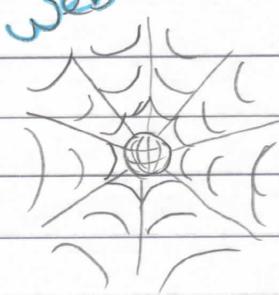


Comprendre le web

Où va apprendre

1. La naissance: web, internet, cloud
2. Les réseaux: IP, nom d'hôte, serveurs, clients, protocoles
3. Ces métiers: Développeurs, UX designers, SI

La naissance



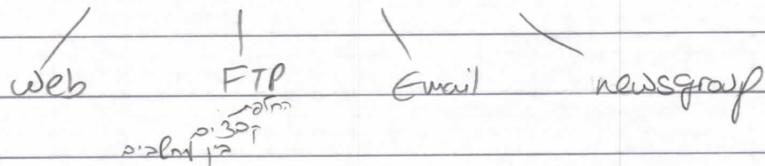
monde sans filtres

à l'origine

World Wide Web → point à

Internet

L'internet et le web ne sont pas la même chose, l'internet a été créé avant le web. L'internet est le réseau qui permet de communiquer entre ordinateurs. L'internet propose des services:



Le web est une partie de l'internet qui utilise les protocoles HTTP et HTTPS pour transférer des pages web.

Cloud



Le cloud est un service qui permet d'accéder à des données et des applications via Internet.

Le cloud inclut des services tels que Gmail, Google Drive et Google Sheets.

Le cloud inclut également des services tels que Google Drive et Google Sheets.

SaaS

Le SaaS (Software as a Service) est un type de logiciel qui est hébergé sur des serveurs dans le cloud.

Histoire

Le web a été créé par Tim Berners-Lee en 1991.

Le protocole HTTP a été créé en 1972.

Tim Berners-Lee a créé "hypertext" pour la première fois. Le web -> 1991

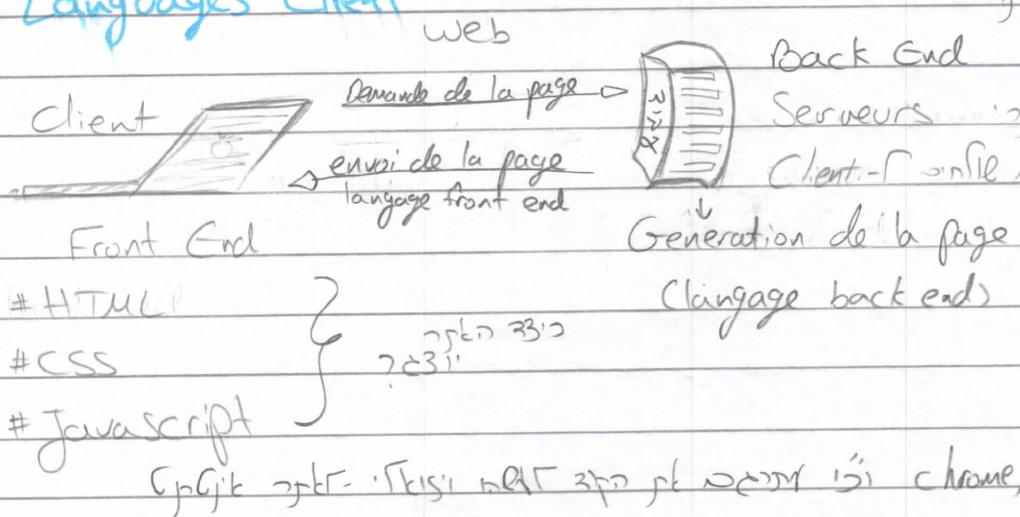
(...en PNG, CSS, HTML) web -> 1993 (Tim Berners-Lee crée W3C -> 1994)

Le web -> 1995 (Le web devient accessible au public)

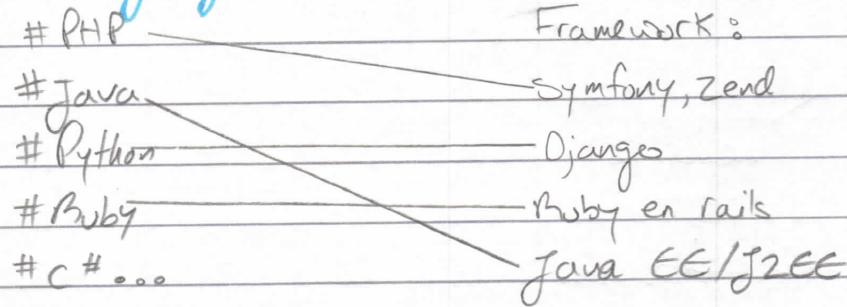
Differents langues de web

- Langues client
- Langues serveur
- Bases de données
- Sites responsives et mobile

Languages Client



Languages Serveurs



Framework : pour faciliter la création de sites web dans ces langages

Framework -> ליתר זמינה לנטור לתוכה של כל שפה. כלומר: אם יש לך שפה מושלמת לאפשרות ליצור מנגנון לשלוט על כל הפעולות בדף

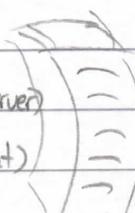


1. demande de la page au serveur

2. Génération de la page (language server)

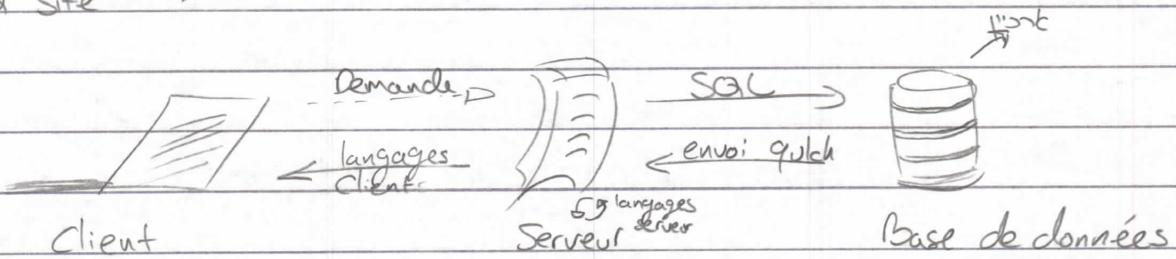
3. Envoi de la page (language client)

לטרא



Découvrez les bases de données

#MySQL
#PostgreSQL }
#SQL Server }
#Oracle }
#SQLite



la base de données est stockée sur un serveur (généralement un serveur différent)

Affichage d'un site Web

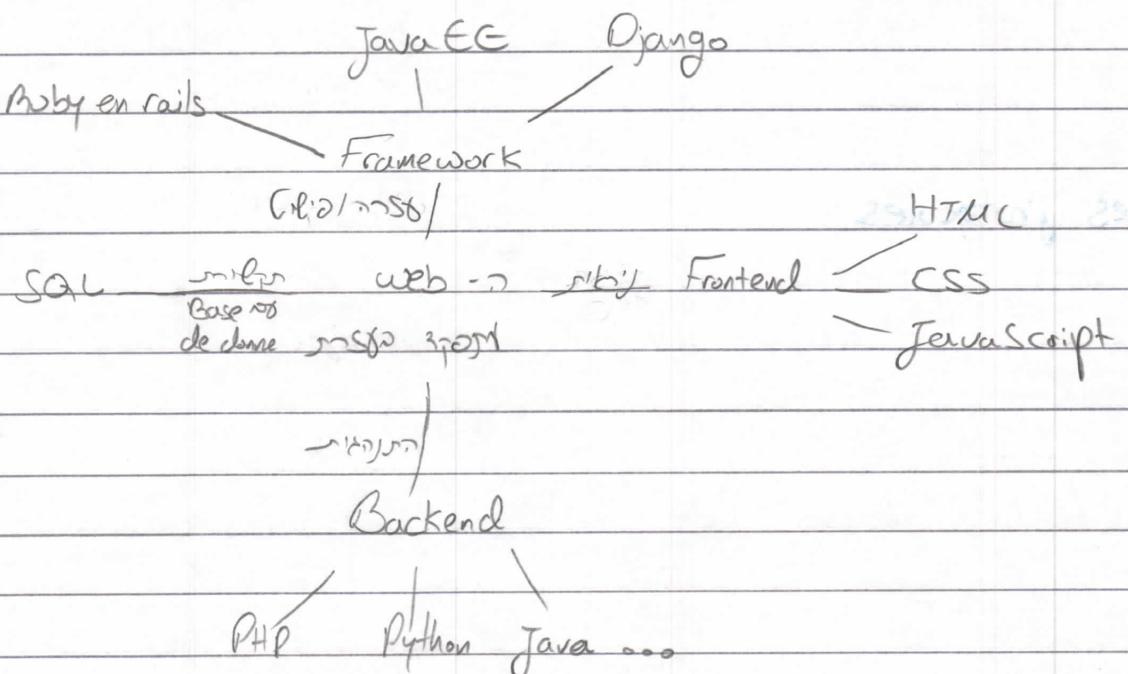
Comment un site web affiche correctement sur un smartphone ?

2 solutions :

une site web responsive : s'adapte à la taille d'écran

une app native : créer une app mobile

En résumé



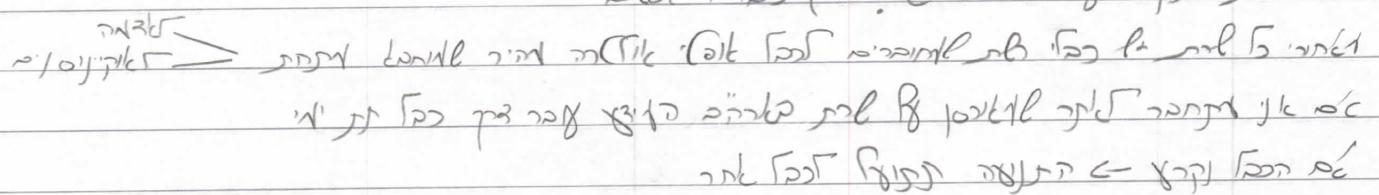
Fonctionnement du réseau

- les serveurs
- Comment on les retrouve
- les Protocoles

Les serveurs

Location: Data Center

machines qui possèdent les sites web et qui les distribuent aux clients



Comment on les retrouve

client: IP serveur: nom de domaine

par nom d'hôtes → google.com → DNS → I.P (205.89.177.26)

DNS = Domain Name Service

le protocole de DNS permet de traduire le nom de domaine → IP

pingTool.org

Les protocoles

Bas niveau: TCP, UDP

Haut niveau:

- HTTP: échanger des pages web entre le client et le serveur
- HTTPS: S = Secure
- FTP: transfert de fichiers
- SMTP: envoi de mail

<http://reborg.com/curl>

HTTP: Générer une adresse web relative au

Metiers de web

5

- Différents professionnels du web
- le métier de développeur
- le métier d'UX design
- technicien informatique

Professionnels du web

Développeur Frontend : HTML, CSS, Javascript

Développeur Backend : PHP, Python, Java, Ruby ...

QA Manager : test du site

Sys Admin : création et config. des serveurs

Fullstack : Frontend + Backend + Sys Admin

UX designers : design et de l'ergonomie des pages web

Jaboscope : podcast qui parle de quotidien des professionnels

Développeur

- code
- teste
- programme
- SQL
- frameworks

Backend : il connaît un langage de prog.

• SQL

• les frameworks

Développeur mobile : développe pour Android ou iOS, il fait le Front et le Back

• SQL

CSS ! HTML 3.1 NT.1

UI/UX 3.2

Javascript 3.3 NT.2

(sysadmin 7.0 NT.1, dev opér 3.1 NT.3, dev opér 3.2)

UX Designer

Assure que la navigation et l'utilisation d'un site/app soit la plus simple, fluide et intuitive.

HCD : Human - centered Design

• je suis un programmeur, je ne suis pas un designer, mais je suis un designer

UI vs UX :

UI : User interface

design graphique, ergonomie, fonctionnel

UX : user experience, design thinking, méthodologie de conception

Product Designer :

UI ! UX sont tous deux très importants dans le développement d'un produit

Technicien Informatique

Assurer le bon fonctionnement du parc informatique, fournir aux utilisateurs des solutions pour pallier les éventuelles pannes / difficultés

... wifi -> Réglage Connexion Internet, ...

HTML5

CSS3

les bases de HTML5

- le fonctionnement des sites webs
- créer votre 1^o page web en HTML
- Organiser votre texte
- créer des liens
- Insérez des images

le fonctionnement des sites webs

- C'est sont les langages informatiques HTML et CSS qui permettent de créer des sites internet
 - le navigateur web fait la traduction entre ces 2 langages, et ce que s'affiche à l'écran
- HTML : est né en 1991. la contenu d'une page web ...liens, menu... etc
- CSS : gérer l'apparence de la page web. 1996

Dreamweaver, Mozilla composer, Microsoft expression web, word!

Sublime text, Notepad++, Brackets, jEdit, PSPad, Context...

caniuse.com

responsive design. lorsque on regarde sur un smartphone ou une tablette ou une autre chose, le site doit se redimensionner pour s'adapter au écran.

Creez votre 1^e page web en HTML

ingrédients:

- text editor: Sublime Text
- testers: Chrome, Firefox, Edge ...

Structure

```
<!DOCTYPE html> → it's an HTML web page that was written in HTML5  
<html>  
  <head>  
    <meta charset="utf-8" /> → sets the encoding to utf-8: 313  
    <title> → sets the title of the page: 313  
  </head>  
  
<body> → contains the content of the page  
</body>  
</html>
```

<Les balises> → <Tags>

invisibles sur l'écran pour les visiteurs de page mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher. il y a 2 types

- les balises en paires: <open> </open>
- orphelines: <image /> il n'y a pas d'attribut

Les Attributs <balise attribut = "valeur">

les attributs sont des tags → le résultat va

Commentaires <!-- la commentaire-->

Organisez votre texte

- Comment rédiger les paragraphes
- Structurer la pages avec titres
- Donner d'importance à certains mots
- Organiser les info. sous forme de listes

Les paragraphes < p >

< p > < /p > *right-align #*

right-align # < p > -> align-left

(les) titres < h1 > → < h6 >

לעתים קוראים ל<h1> כtitel ו<h2> כheader. במתוך היררכיה, מושך <h1> ו<h2> ו<h3> וכו' מ*<h6>. מושך מ*<h1> וכו' מ*<h6>css. מושך מ*<h1> וכו' מ*<h6>css.

Mise En Valeur

< em > → *italic* } Important

< strong > → *bold* }

important -> bold italic / Bold -> *important* & < em > ! < strong > : important < em > : important < em > : important < strong > : important < strong > : important

< marks > → *mark*

important < mark > < marks > ->

HTML : le fond (contenu, logique des éléments)

CSS : la forme (apparence)

(les) paragraphes (continue)

< p > -> *center* -> right-align

< p > -> *left-align*

left-align

< p > -> *right-align*

css -> right-align

Les listes

- liste non ordonnée
- liste ordonnée —
 - 1 *Fraises*
 - 2 *Tomates*
 - 3 *bread*

- list de définitions (rare)

liste non ordonnée :

- Bread
- Butter
- Milk

Bread → 1st word
Butter → 2nd word
Milk → 3rd word

 Bread

 Butter

 Milk

: אֶלְגָּמָן. יְמִינָה אַרְבָּתָה לְבָנָה גַּמְלָה גַּמְלָה

• Drinks

- Vegetables:
 - Apple
 - pineapple
 - Celery
- wash it
 - Dry it
 - with napkin
 - really dry

• Main course

liste ordonnée

1. Je me lève
2. Je mange
3. Je m'endors

 Je me lève

 Je mange

 Je m'endors

Définition

<dl> *term* *definition*

<dt> term *definition*

<dd> definition *definition*

טֵבַע אֶתְמָלֵת בְּנֵי עֲמָקָם כָּלְבָּד כָּלְבָּד כָּלְבָּד כָּלְבָּד

Créer des liens

< a href="x" >

- un lien vers :
 - un autre site
 - une autre page de mon site
 - une ancre
- special cases
 - email address
 - Download a file

useful attributes

< a href="x" > ...

un autre site

x = the site complet address → lien absolu (<http://...>)

& si dans l'adresse absolu il y a un & il faut le remplacer par &

une autre page de mon site

לפיה של דף אחר בדף הנוכחי
הכתובת של דף אחר מושב בפיה של דף הנוכחי
לפיה של דף אחר מושב בפיה של דף הנוכחי

- x = the file name → lien relatif (name.html)

link מהו נושא של דף זה בפיה של דף אחר

sous dossier:

x = foldername/filename

dossier parent:

x = ../filename

une ancre

הברוך ישב ימינו ורבות שרים ורבים ישבו בזיכר ישבו בזיכר
לפיה של דף אחד שמהו לינק לenthal [link](#) id="mon-ancre" id="mon-ancre"

< h2 id="mon-ancre" > Titre </h2 >

x = #name of the id I gave

x = #the file name #name of the ID

link בפיה של ancre -> ok ok
link בפיה של location -> ok

special cases

email: x = mailto:myemailaddress@email.com

file download: x = filename.exe, zip, ...

useful attributes

title: show the input of title when I hover with the mouse near the link

target = "blank": link יפתח כפנוי בחלון חדש, link יפתח כפנוי

Insérez des images

- Les différents formats : JPEG, PNG, GIF
- Insérez un image
- Les figures

Ces différents formats

- JPEG : jpg / .jpeg => photos ce format est conçu pour réduire le poids des photos
- PNG : 2 avantages - 1. peut être transparent => pour tous les illustrations, logos, dessins
2. n'affecte pas la qualité de l'image
- GIF : limité à 256 couleurs => animations

Gif => Peut 256 couleurs max 16 bits => Png => Jpeg =>

Jpeg => 8 bits par pixel Png => 24 bits par pixel Jpeg => 24 bits par pixel

(16 bits) Png24 => 256 couleurs (max 256) Png8 => 256 couleurs max 8 bits par pixel

Bmp => 24 bits (16 bits + 1 bit alpha) bmp => Png. Il existe aussi des formats comme gif89 qui sont des versions améliorées de gif.

Insérez un image

2 att obligatoires :

• src = source of the image. absolu : http://... .com ou relatif : images/fleur.png

• alt = text alternatif. a MUST. un court text qui décrit l'image

autre att : title = facultatif. the same purpose as in the links

1: resize the photo (can use Resizeimage.net)

2: créer un lien

!>

Ces Figures <figure>

!> les 3 types de figures : images, captions et figures

<figures> <p>-> 1st figure 1st fig /> <figures>-> 2nd figure 2nd fig

<figcaption> ... </figcaption> -> 1st figure 1st fig

</figures>

Mettez en forme votre page avec CSS

- mettez en place le css
- Formatez du texte
- Ajoutez de la couleur et un fond
- Créez des bordures et des ombres
- Créez des apparences dynamique

Mettez en place le CSS

css : ça me permet de choisir - la couleur de texte - la police - la taille
- les bordures (border) - le fond (background)
mise en page

there isn't a web browser that support all the fitchers of CSS to find out if it support a function of CSS : www.canuse.com

Where I put my CSS code?

- ✓ # a separate .css file → the best practice. that a link for it is planted in the <head> after <meta>. the form: <link rel="stylesheet" href="filename.css">
- ✗ # inside the <head> → first, no. on met le CSS code direct dans une balise <style> that comes after the <meta> and should be closed </style>
- XXX # Direct dans les balises <body> avec le att. <style="x:y;">

Applique un style

- selectionner une balise

1 p → le nom de <balise> dont on veut modifier l'apparence

2 {

3 propriété: valeur; → je peux indiquer combien de propriétés que je veux

4 }

5 } → je veux appliquer la propriété sur p et h1 aussi

{

propriété: valeur

}

Comments /* comment */

/* */

/* block */ /* inline */ /* block */ /* inline */ /* block */ /* inline */

• class et id

class -> tag ou élément avec une classe CSS

<balise class="identifier"> ← html -> la balise a une classe .identifier ← valeur de la classe

CSS -> style associé à l'élément

{ prop = valeur }

va être modifier <x> class="tag" tag -> style associé à l'élément

id ≈ class → le id fonctionne exactement de la même manière comme un class

mais il est unique : il ne peut être utilisé qu'une fois dans le code html

class -> élément CSS ->

id -> élément CSS -> id -> CSS -> JS -> JavaScript -> valeur de l'id -> autre élément

(HTML, CSS, JS)

#idname

... à l'élément CSS associé à l'élément HTML -> style associé à l'élément

{ prop = valeur; }

• Les balises universelles <div>

→ <div> <p> -> Région de l'élément avec la classe class="block" ou "block"

→ <div> <div> -> Région de l'élément avec la classe class="block" ou "block"

• : inline. une <balise> que l'on place au sein d'une <p> </p>

pour sélectionner certains mots uniquement. sont de la même famille.

• <div> </div> : Block. qui entoure un block de text. <p> </p> sont de la même famille.

elles créent un nouveau "block"

insert → text box

• Les sélecteurs

* : sélecteur universel : sélectionne toutes les <balises> sans exception

h1 em : toutes les situées dans <h1>

h1

h3 + p : sélectionne la première <p> située après <h3>

h3

a[title] : sélectionne tous les liens <a> qui possèdent un att. title

balise[att.]

a[title="cliquez ici"] : idem, mais avec un valeur spécifique

balise[att="valeur"]

a[title*="ici"] : l'att doit contenir dans sa valeur la mot "ici" par exemple

* tout élément

E tout élément de type E

E[foo] tout élément E portant l'attribut "foo"

E[foo~="bar"] tout élément E portant l'attribut "foo" et dont la valeur de cet attribut est exactement "bar"

E[foo\$="bar"] tout élément E dont la valeur de l'attribut "foo" commence exactement par la chaîne "bar"

E[foo*="bar"] tout élément E dont la valeur de l'attribut "foo" contient la sous-chaîne "bar"

E[lang|="en"] tout élément E dont l'attribut "lang" est une liste de valeurs séparées par des tirets et commençant (à gauche) par "en"

E:root un élément E, racine du document

E:nth-child(n) un élément E qui est le n-ième enfant de son parent

E:nth-last-child(n) un élément E qui est le n-ième enfant de son parent en comptant depuis le dernier enfant

E:nth-last-of-type(n) un élément E qui est le n-ième enfant de son parent et de ce type

E:first-child un élément E, premier enfant de son parent

E:last-child un élément E, dernier enfant de son parent

E:first-of-type un élément E, premier enfant de son type

E:last-of-type un élément E, dernier enfant de son type

E:only-child un élément E, seul enfant de son parent

E:only-of-type un élément E, seul enfant de son type

E:empty un élément E qui n'a aucun enfant (y compris noeuds textuels purs)

E:link un élément E qui est la source d'un lien dont la cible n'a pas encore été visitée (:link) ou a déjà été visitée (:visited)

E:active un élément E pendant certaines actions de l'utilisateur

E:hover un élément E qui est la cible de l'URL d'origine contenant lui-même un fragment identifiant.

E:lang(c) un élément E dont le langage (humain) est c (le langage du document spécifie comment le langage humain est déterminé)

E:enabled un élément d'interface utilisateur E qui est actif ou inactif.

E:disabled un élément d'interface utilisateur E qui est coché ou dont l'état est indéterminé (par exemple un bouton-radio ou une case à cocher)

E:contains("foo") un élément E dont le contenu concaténé contient la sous-chaîne "foo"

E::first-line la première ligne formatée d'un élément E

E::first-letter le premier caractère formaté d'un élément E

E::selection la partie d'un élément E qui est actuellement sélectionné/mise en exergue par l'utilisateur

E::before le contenu généré avant un élément E

E::after le contenu généré après un élément E

E:warning Uniquement en HTML. Identique à E[class~="warning"].

E#myid un élément E dont l'ID est égal à "myid".

E:not(s) un élément E qui n'est pas représenté par le sélecteur simple s

E_F un élément F qui est le descendant d'un élément E

E > F un élément F qui est le fils d'un élément E

E + F un élément F immédiatement précédé par un élément E

E ~ F un élément F précédé par un élément E

www.w3schools.com/cssref/trysel.asp

Formatez du texte

- La taille
- Les flottants

- La police

- I, I, I

- L'alignement

• La taille font-size

avec la prop. font-size, on peut indiquer une:

#taille absolute: en px, cm, mm... très précise.

#taille relative: en x, em, ex, mots

em: 1em c'est la taille normal

ex: la même principe comme em mais le ex est plus petit

mots: xx-small → x-small → small → medium → large → x-large → xx-large

medium=1em

• La police font-family

font-family: serif, sans-serif, monospace, etc.

font-family: Arial, Helvetica, sans-serif; ou Georgia, Times, serif;

font-family: "Times New Roman", "Times New Roman", serif;

serif / sans serif: A à serif A à sans serif

safe font: liste de polices qui fonctionnent bien sur la plupart des navigateurs

Arial; Arial Black; Comic Sans MS; Courier new; Georgia; Impact

Times New Roman; Trebuchet MS; Verdana

#Si le nom comporte des espaces comme Arial Black c'est mieux de l'écrire "Arial Black"

Font by demande @font-face

font.squirrel.com ou dafont.com

en CSS3: font-family: "Myriad Pro", sans-serif; font-weight: bold; font-style: italic;

download < webfont kit

Font file types:

✓ WOFF: nouveau format conçu pour le web. fonctionne sur IE9 et tous les autres navigateurs

TTF: used by the operating system. fonctionne - - - - -

EOT: only for internet explorer

SVG: the only one that recognise by the Ipad and iPhone

OFT: ne fonctionne pas sur Internet Explorer

Définition d'une nouvelle police

@font-face {

font-family: fontname

 src: url("fontname.woff") format('woff')
 url("fontname.ttf") format('truetype') }
 url("fontname.eot") format('eot') }
 url("fontname.svg") format('svg') }
 css -> שורש פולש נספחים כוונת מילוי דוחה ורנינה
 המקורה בדוחה מילוי דוחה ורנינה
 המקורה בדוחה מילוי דוחה ורנינה

url('fonts/fontname.ext')

url -> סט פונטים הנדרש עיבוד ערך

רפל רילוקט ב"*.eot ?#iefix"-ר מvolt squirrel-ר CSS שיפת קומל ערך
רפל רילוקט ב"*.eot ?#iefix"-ר מvolt squirrel CSS שיפת קומל ערך
רפל רילוקט ב"*.eot ?#iefix"-ר מvolt squirrel CSS שיפת קומל ערך

Italic, Bold, underline ...

la `` était pas fait pour mettre le texte en italique et `` non plus

#italic → font-style: italic "c font-style: oblique
let's say that I have used an `` to mark as important but I don't want it to be italic:
`font-style: normal`

#bold → font-weight: bold

`font-weight: normal` ← (מגדן (אלא נורם))

#underline → text-decoration: underline

`#overline` → text-decoration: overline

#line-through → text-decoration: line-through

`#blink` → text-decoration: blink !()

L'alignement text-align

left: this is the default

right: ימין

justify: it takes all possible space without leaving white space in the end of the lines.

inline (בצטראט, `<hr>, <div>, <p>` block בודניך פג' יז Justify first ok #!)

Ces flottants

ce CSS permet de faire flotter un élément autour du texte.

float: - left - right ← זוהי 2 פלאים

`<block>` פלא `<inline>` פלא

I must put the floating element first in the html code. if second - it won't work!

folk `
` זוהי פלא יק, זה פלא מילא את כל הפלג הזה (copy) פלא מילא את כל הפלג הזה
: "131 float-left 131" (copy) פלא מילא את כל הפלג הזה ← left ← left ← left ← left

clear: left; if the element is floating to the left

right; if it float to the right

both; if it float to the right/left

← -- מילא שפה פלא מילא פלא מילא ↗

73P מילא יוג< p class="..."> זה מילא יוג html → זה מילא יוג

Ajoutez de la couleur et un fond

- couleur de texte
- couleur de fond
- héritage
- image de fond
- transparence

Couleur de texte color

Il y a quel façon de choisir une couleur:

#nom: le plus simple, par contre il y a que 16 : -white -black -silver -gray
-red -marron -green -lime -yellow -olive -blue -navy

-fuchsia -purple -aqua -teal

#hexadecimale: #RRGGBB the values are 0-a and A-F

#RGB: rgb(red value, green value, blue value)

www.colorpicker.com

RGB : 163:30:70 (25% black + 50% red + 25% green)

Couleur de fond background-color

body { background-color: ; } // le fond de la page

color: ; } // le texte de la page

<mark> sera noir, mais si je le place dans une balise de style, il sera vert

Héritage

La règle d'héritage est la suivante: Si deux éléments ont la même propriété, alors celle qui est définie dans l'élément parent sera utilisée. Cela signifie que les propriétés qui se trouvent à l'intérieur d'une autre balise "héritent" de ses propriétés.

Le CSS décide qui c'est la déclaration la plus précise qui l'emporte.

Exemple: `strong { font-size: 1.5em; } p { font-size: 1.3em; }` Si j'écris `p`, alors la taille de l'élément sera de 1.3em, pas 1.5em. Cela signifie que l'élément parent a une taille de 1.3em et que l'élément enfant a une taille de 1.5em.

Image de fond background-image

La règle d'héritage pour la propriété `background-color` est la suivante: Si deux éléments ont la même propriété, alors celle qui est définie dans l'élément parent sera utilisée. Cela signifie que les propriétés qui se trouvent à l'intérieur d'une autre balise "héritent" de ses propriétés.

Par exemple, si j'écris `background-color: url('filename.jpg');` dans un élément parent, alors l'élément enfant héritera de cette propriété. Cela signifie que si j'écris `background-color: black;` dans l'élément enfant, alors l'élément parent aura toujours la couleur de fond spécifiée dans l'élément parent.

Controlling image behavior

repetition background-repeat

- no repeat: one photo without repetition

→ 15 (3) 11 "no repeat" it's not repeat

- repeat: Default. le fond sera répété en mosaïque

- repeat-x: will be repeated only in the first line, horizontally.

↔ 3 7 15 225

- repeat-y: - 11 - 11 - 11 - 11 - 11 column, vertically.

↑ 225

fixation background-attachment

- fixed: the text glisse over the pic. when I scroll → only the text scroll

- scroll: the default. when I scroll → both the pic and the text scroll

position background-position

in order to see its effect we must set background-repeat: no-repeat;

{ background-position: px px }

left top : it will be position Xpx from the top and Ypx from left

there is an option to describe in english:

:top; :bottom; :left; :center; :right;

if it's possible to combine the words like: top right; bottom left; etc...

⑤ background: super-propriétés

background: url("x.png") no-repeat fixed

→ 3 27 ← background ← 3rd propriété sur seconde 5 16 15 16

1,2,3 plusieurs images de fond

since CSS3 it's possible to attribute multiple image to one element

to do this I need to separate the declaration by , (comma):

Background: url("x.png") no-repeat fixed, url("y.jpg") top right no-repeat;

→ 16 20 → 18 19 → 17 18 → 19 20

a trois méthodes pour 10 11 12 13 14 15 16 17 18 19 20

La transparence opacity

0-1: 0 = transparent

1 = opaque

when you write opacity: 0; <sp> R F G B <sp> when you write opacity: 0.5; <sp> R F G B

<sp> R F G B when you write opacity: 0.5; <sp> R F G B

rgba

if you write places: red, green, blue, alpha → pour 4 valeurs

like as in 15 16 or 17 18 rgba(255, 255, 255, 0.5)

Creez : Bordures et Ombres

o Bordures standard

o Bordures arrondies

o Les Ombres

Bordures

Standard

border:

css propose some properties for controlling borders:

border-color , border-width , border-style etc..

instead of using all of them we will use only one : border

border: super-propriété . on peut utiliser jusqu'à 3 valeurs

largeur: in px couleur type: there are 10

:none

:solid

:groove

:dotted

:ridge

:dashed

:inset ~~right Groove~~ } 3D effect

:double

:outset ~~left Groove~~

position

border-top: ... ; border-bottom: ... ; border-left: ... ; border-right: ... ;

Rounded

border-radius: px

border-radius: px px px px

Ombres

Ombres des boîtes

box-shadow:

box-shadow: px px px color (inset); facultatif: px px

soft hard transition blur

transition: faible normal élevé

du texte

text-shadow

box-shadow: px px px color (soft, hard, drop shadow) room

Apparences

- Au survol

Dynamiques

- Au click

- lien qui a déjà été consulté

les pseudo-formats

(element):hover{...} survol

לינק עבוי מודפס

Apparence

click

(element):active

לינק מודפס

(element):visited

לינק שכבר נזקק

~~xs3~~ לינק מסומן כבוקס

Dynamique

היה בזקק

(element):focus

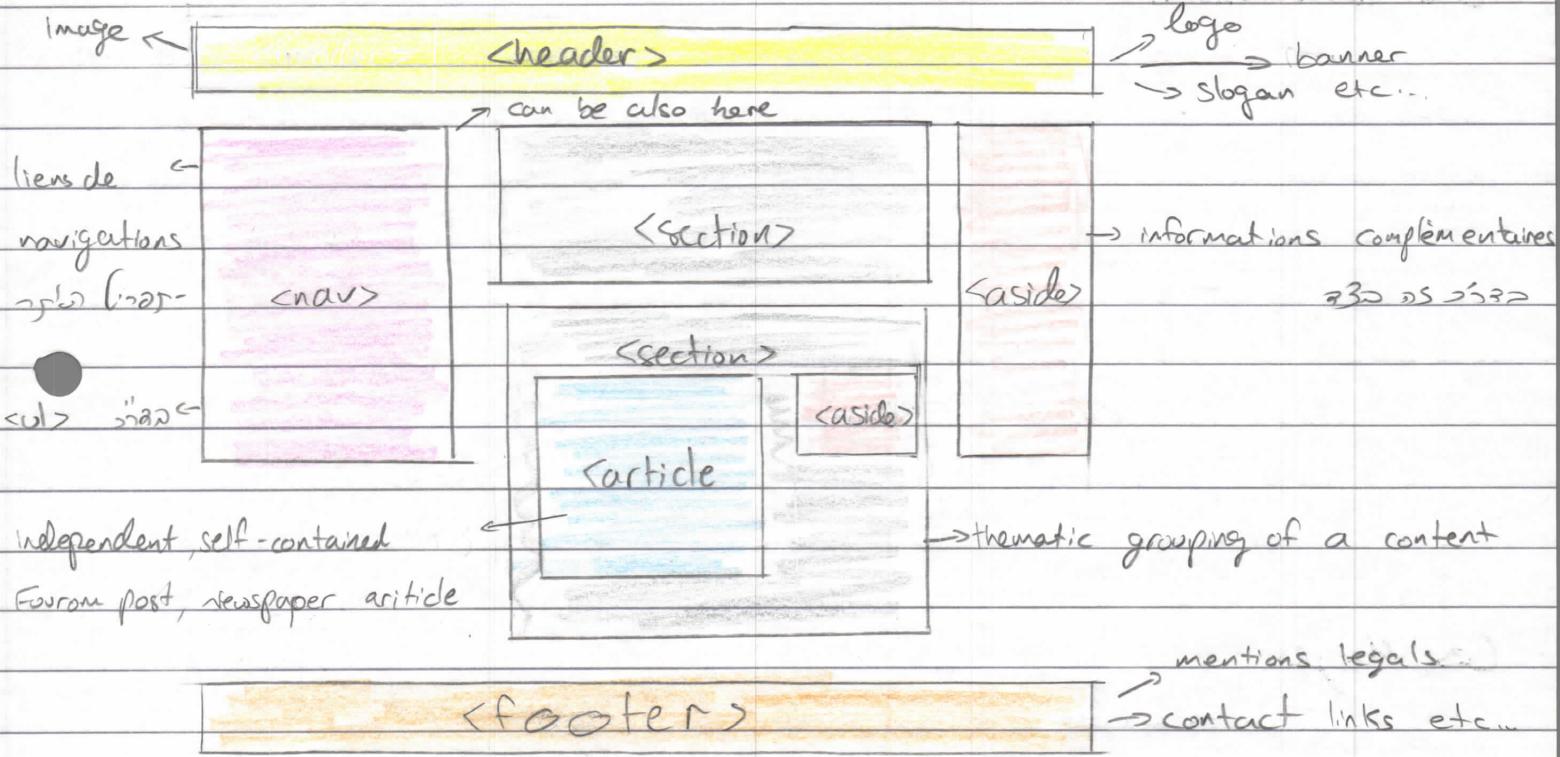
tab לינק מודפס, מודפס

לינק מסומן כבוקס

Mettez en page votre Site

- Structurez le page
- les modèle des boîtes
- ilise en page: Flexbox
- Mise en page: autres techniques
- TP: créez un site pas à pas

Structurez le page



Les modèle des boîtes

- type block et inline
 - les dimensions
 - les marges
 - quand ça dépasse
- blocks: block-level elements (div, section, etc.) have their own rectangular box with width and height. They can overlap each other.
- margins: margins are gaps between block-level elements. They do not overlap.
- type block et inline

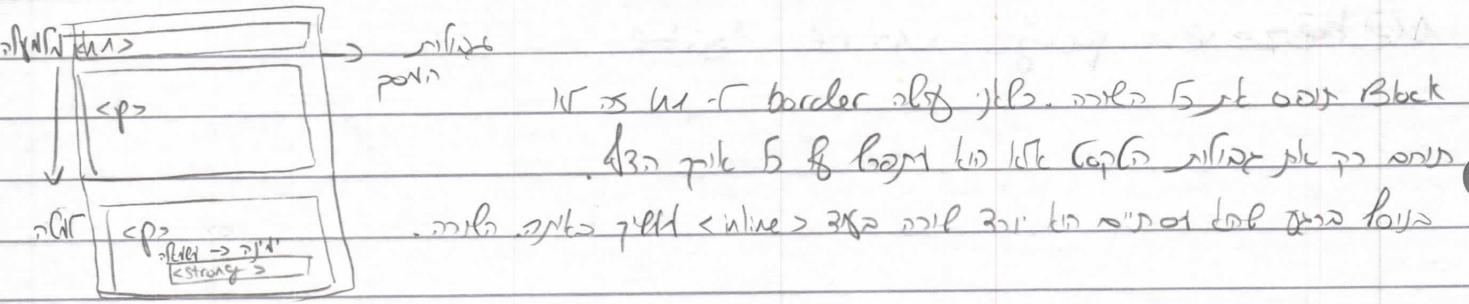
block

inline

<header> <nav> <section> <articles>
<aside> <p> <h1 ...> <div> <figures>
<col> <cl> <nav>

 <mark>

Border: border of block, margin of block



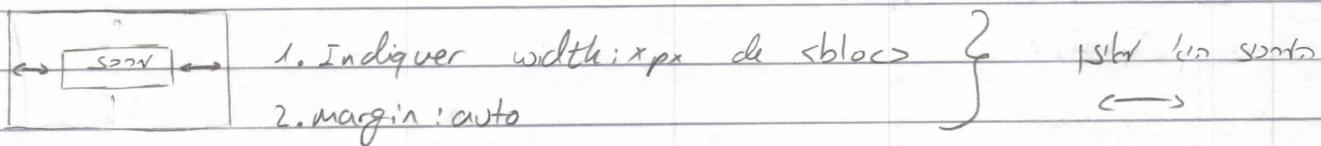
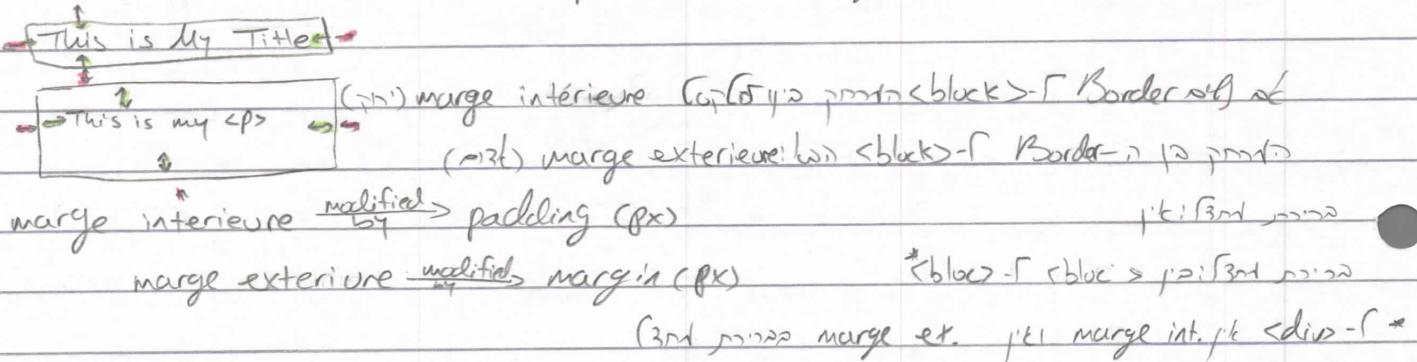
Les Dimensions

width: en px ou %. ↗ par défaut un bloc prend 100% de la largeur disponible
height: en px ou %. (fixe) ↗ 100px des blocs
 ↗ par créer un design qui s'adapte auto. à la rés. d'écran du visiteur
min-width: prevent the value of width to become smaller than what I defined



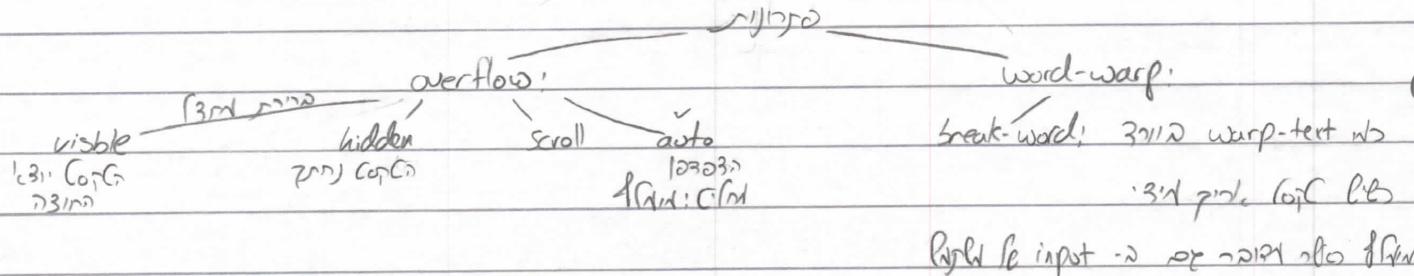
Les Marges

Tous les éléments les blocs possèdent des marges



ça dépasse

border溢出 (border overflow) : si l'élément dépasse le cadre de son parent, il peut dépasser



Right: input → position absolute after flex

Flexbox

o un conteneur, des éléments

o soyez flex

o Alignez-les!

o Répartir plusieurs lignes

o Rappel à l'ordre

o Grosse marge

Un conteneur

HTML

```
<div id="conteneur">  
  | element 1 | element 2 | element 3 |  
  <div class="e1>element</div>  
  <div class="e2>element</div>  
  <div class="e3>element</div>  
</div>
```

css

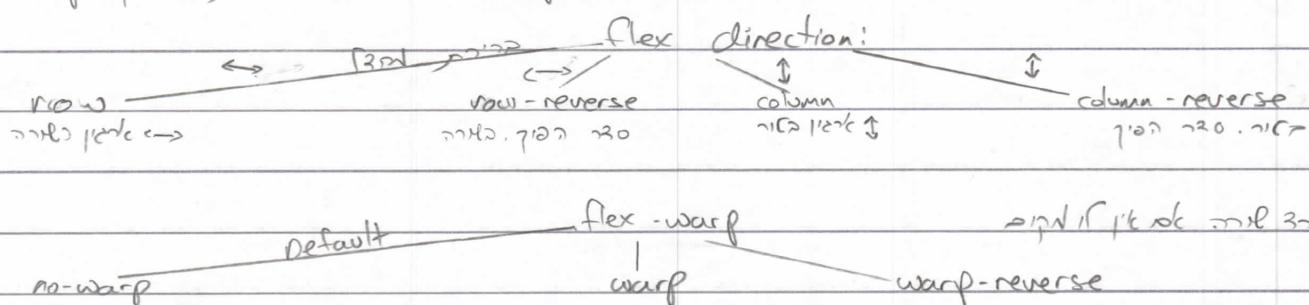
```
#conteneur {  
  display: flex;  
  flex-direction: ;  
}
```

Flex

display: flex;

display: flex; \Rightarrow les blocs se place côté à côté

(冒 (5) □□□



Alignez-les !

Axe Principal

justify-content:

flex-start

flex-end

center

space-between

space-around

Default -> flex-start

justify

space

Le n° 1 est height et n° 3 est le n° 2 (column) si l'élément a une hauteur fixe (column).

Axe Secondaire :

align-items:

stretch

baseline

flex-start

flex-end

center

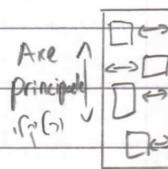
align-top

flex-start

align-bottom

flex-end

align-center



\leftrightarrow Axe Secondaire

Aligner 1 seul élément: align-self
élément:nth-child(2) => on prend le 2^e bloc élément/
align-self: flex-end (for example)

plusieurs lignes

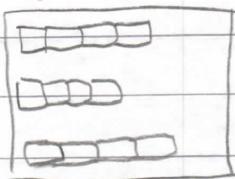
#container {

display: flex;

flex-wrap: wrap; /> si on a 100% on l'écrit à la fin

align-content: flex-start\flex-end\center\space-between\space-around\stretch

align-content



→ ligne 1

→ ligne 2

→ ligne 3

Ordre

order:

on peut changer l'ordre des éléments sans toucher le code html

.element:nth-child(1){ 1 = 1er élément } ordre 1
order: 3; 3 = 3^e élément / 2 = 2^e élément }

Taille

flex:

flex-shrink: 1 => 1/3 de la taille

flex-grow: 1 => 1/3 de la taille

flexion autorise nos éléments à occuper plus ou moins d'espace restant

1^e valve : flex shrink 2^e : flex-grow

3^e : flex-basis

flex-basis: %, how much of the screen size it will occupy

Where

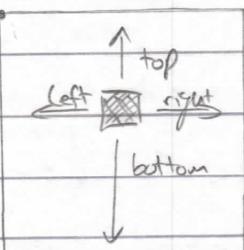
pixel

0,0

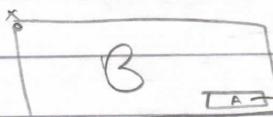
הזיהוי ביחס לנקודה (0,0)

(0,0)

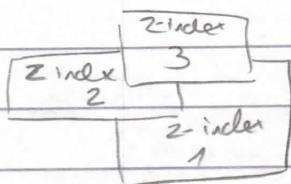
%



position: absolute ביחס לנקודה (0,0)



z-index



הערך הגבוה יותר הוא באחור

הערך נמוך יותר הוא קדימה

הערך נמוך יותר הוא קדימה

הערך הגבוה יותר הוא באחור, נמוך יותר הוא קדימה

Relatif

הערך הגבוה יותר הוא באחור, נמוך יותר הוא קדימה. הערך (0,0) הוא מוקטן.

Hankley Hello How Are **you**? I am OK.

css: strong { position: relative }

הערך הגבוה יותר הוא באחור, נמוך יותר הוא קדימה.

הערך נמוך יותר הוא קדימה, הגבוה יותר הוא באחור. הערך left הוא מוקטן.

הערך נמוך יותר הוא קדימה, הגבוה יותר הוא באחור.

left: 10px → 1x ↑
border-left

Creez une site

383 303

- Maquettage du design
- Mettre en forme en CSS

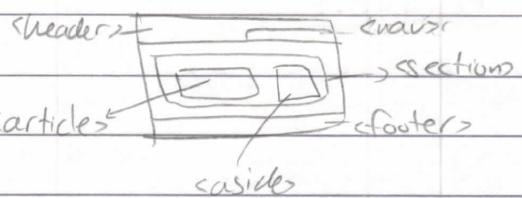
- Organiser le contenu en HTML
- La validité
- Le code final

La Maquette

What is the objective? the theme? what is it talking about?

Design! - moi même

- une site qui aide (htl5up.net)



HTML

- Writing all the blocks names in html
- putting everything inside <div id="main"> ⇒ pour organiser les blocs et mettre tout dans un seul élément

CSS

- ① Police personnalisées
- ② Définition des styles principales de la page (largeur de site, fond, couleur de texte...)
- ③ En-tête et liens de navigation (header)
- ④ Bannière
- ⑤ Section principale du corps de page
- ⑥ footer

polices perso

Font Squirrel → Generator
css 317 files sont générés pour la police

FontSquirrel → Generator

css 317 files sont générés pour la police

Style de la page

#main {
text-transform: uppercase;

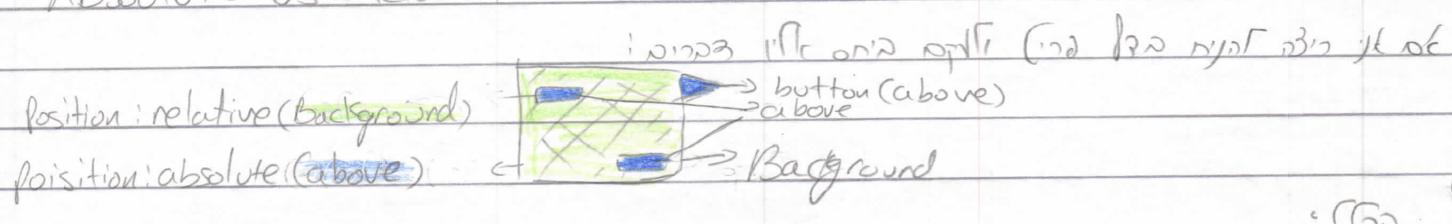
En tête + lieux de navigation

Logo: `HTML-CSS Background`
`nav ul {list-style-type: none;}` \rightarrow `list-style-type: none;`
`display: flex;` \rightarrow `flex-direction: column;`
`nav li {margin-right: 30px;}` \rightarrow `margin-right: 30px;`

La bannière

`background-image: url('');` \rightarrow CSS `url()` \rightarrow `<div>` \rightarrow `background-image: url('');`

Absolute vs Relatif



(Fixed / sticky / fixed / in relation to parent (P)) \rightarrow block \rightarrow position: absolute \rightarrow parent \rightarrow position: absolute \rightarrow style

Le Corps

absolu - relatif \rightarrow position: relative \rightarrow position: absolute

Liste de propriété CSS

position: absolute

Test

<http://validator.w3.org>

proper \rightarrow style \rightarrow `img alt="Image alt text"`
`` \rightarrow `
` \rightarrow `
` \rightarrow `
`
`<alt>` \rightarrow `img alt="Image alt text"`
`&amp` \rightarrow `img alt="Image alt text" <img alt="Image alt text"`

3 Pillars for Building a Website

1

- 1) Responsive design: fluid layouts, media queries, responsive images
correct units, desktop first - mobile first
- 2) Maintainable\Scalable code: clean, easy to understand, organized, reusable.
- 3) Web performance: faster, smaller in size \Rightarrow less http, less code, less images, compress images, use CSS preprocessor, compress code

Css Behind the Scenes

parsing CSS | resolving CSS conflict declaration

process final CSS values

Resolving CSS Conflicts

CSS sources: Author (the developer), user, browser

Cascade!

importance (weight) \rightarrow specificity \rightarrow source order

- 1) user !important
- 2) author !important
- 3) Author
- 4) User
- 5) Browser

1) inline styles

2) id

3) class, pseudo-class, attr

4) Element, pseudo-ele

\downarrow
the last declaration
in the code will
override all others
and will be applied

\downarrow
cascade value

cascade value:

→ 100% of 25% of 100%

if something doesn't work the way I expected, I should consider the option that it's a specificity problem

process values

everything will be converted to pixels at the end

1) Specified value: the default in case there isn't cascade value

66%

2) Computed value: converting relative values to absolute ($1.5\text{rem} \Leftrightarrow 24\text{px}$)

66%

3) Used value: final calculations

184.8px (66% of 280px)

4) Actual value: $\frac{\text{final}}{\text{specified}}$ round(value) 185px

All in one

Declared value \rightarrow Cascaded value \rightarrow Specified value \rightarrow Computed value \rightarrow Used value
 $\frac{\text{actual}}{\text{specified}}$ value
 $\frac{\text{actual}}{\text{cascaded}}$ value
 $\frac{\text{actual}}{\text{declared}}$ value

Default \leftarrow value if not in CSS property 5, value \rightarrow if it is or

Each CSS property has an initial value in case there is no cascaded value

computed values! From relative → absolute

When using length, the computed value uses its parent width as a reference
headers
width: 100px 3

header-child 2

padding: 10% 3 10% of what?

→ of 100px

use font of em: can serve for font or for length, em use the parent as a ref
as rem: use the root font size as a ref html { selector

em(length): use the font-size of the ^{current element} parent as a ref

Why to have the font size as a ref? it's a powerful tool in responsive design
just by changing the font-size, we change all the elements that user em/rem

viewport ref
vh: 1vh → 1 percent of the viewport height
vw - 1 - width

in inheritance it's not the specified value that being inherited (66%, 1em etc...)

it's the actual value (26px for ex.)

padding is not inherited

rem is more easy to use since we always know the ref (root) but with em
we need to ask "who is the parent?" and make calculation accordingly
Rem doesn't apply to IE9 and less

Good Practice

set the font size in the html as percentage (the browser default is 10px)
all the measurement are in rem

The Box Model

if I don't specify the height/width → the content of the box will determine
its size. if I specify width - the total width will be
right border + right padding + content specified width + left padding + left border
to bypass this behavior → box-sizing: border-box

Box types

2

inline

box-level

inline-block

Box-level

display: block / flex / table / list-item

elements formatted as blocks, vertically one after another

100% parent wide, will occupy as much space as possible
default property for divs and paragraph

inline

display: inline

content is distributed in lines

no line breaks

no height and width

occupies only the content's space

padding/margins are only horizontal (left/right)

sits inside their parent block element

inline-block

display: inline-block

a mix of the two

occupies only content space

no line breaks

work as a block-level on the inside and inline on the outside

Positioning

Normal flow: Default, ele laid out according to their source order {position: relative}

Floating: the ele is taken out of the normal flow, text and inline ele will wrap around it, the container won't adjust its height to the ele {float: right/left} the ele is pushed to the left/right as far as possible → solution: clear

Absolute: the ele is removed from the normal flow, there is no impact on surrounding content/element and this is the difference between absolute and float. we use top/bottom/... to offset the ele from its relatively positioned container {position: absolute/fixed}. If it overlaps an element I can control who is on-front/behind with z-index