



miam

Web

Janvier-avril 2023

JUNIA

HEI · ISEN · ISA

Grande
école
d'ingénieurs

Un petit recap

Cours 1

Je crée un fichier Html basic, un fichier js.

Je sais importer mon js dans mon html à l'aide d'une balise script

Je sais faire des fonctions en javascript

Mon js me permet de faire des actions et modifier de façon asynchrone le html pour exécuter des actions (changer une couleur)

Cours 2

Notion de DOM : une arborescence du site avec des balises / nodes.

C'est ce que j'ai fais dans mon propre html

J'apprend à parcourir le DOM pour récupérer des informations

Je vois les requêtes qui partent de mon front vers une api.

J'inspecte cette requête pour la reproduire

Je la reproduit à l'aide de XMLHttpRequest

Cours 3

Notion orientée Objet. On commence par voir comment ça fonctionne puis le sucre syntaxique de la class

Héritage

On change intègre les fonctions d'appel à l'api dans des classes bien propres avec de l'héritage. Je peux me re-servir de ces classes pour récupérer des annonces d'une autre catégorie par exemple avec très peu de code.

Cours 4

On se crée un serveur basic en node

On ordonne les route qu'il accepte

on traite un envoie d'annonce depuis notre script leboncoin et on enregistre la data dans un fichier à l'aide d'un service.

Next step

On se sert d'une BDD plutôt qu'un fichier

On Travail avec des frameworks qui font le travail plus proprement et plus facilement

On se fait un petit projet global avec un back qui peut récupérer des annonces et fournir la data à notre front qui permet d'afficher des annonces de plusieurs site.

On a réalisé un projet complet d'agrégateur d'annonces.

Entre deux : un exercice plus simple en reprenant les bases :

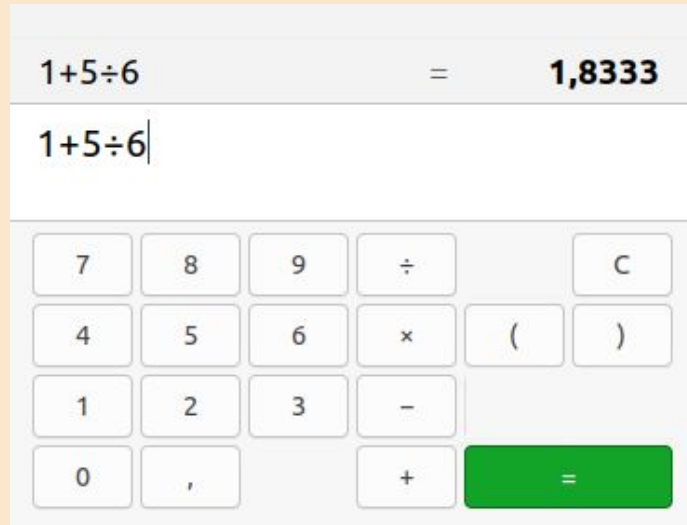
Le but : Une calculatrice ++

- Créer une page (Html + css) lié à un javascript pour réaliser une calculatrice
- Envoyer des événement à un serveur local :
 - Si opération valide : temps prit pour réaliser l'opération (c'est à dire le temps entre le premier caractère écrit et la demande de calcul)
 - Si opération invalide (par exemple l'utilisateur tape 1++2) un événement d'erreur
- Lors de ces événements le serveur renvoi un rapport global :
 - Si événement valide : %de calculs plus rapide que moi, moyenne de temps de calcul.
 - Si invalide : Nombre d'événement invalide, date de la dernière précédente erreure
- Le front affiche ces rapports

Partie 1 : Le front pure html / css

Réaliser la partie purement html / css de la calculatrice avec

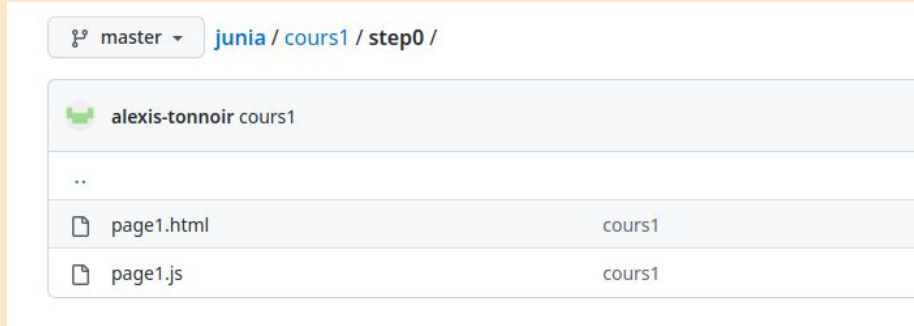
- les 9 chiffres et la virgule/le point
 - les opérations $+$, $-$, $*$, $/$
 - les parenthèses
 - un bouton pour effacer la dernière opération
 - une zone où apparaîtra ce que je tape
 - une zone avec le résultat
- ça doit ressembler à ceci



Partie 2 : Le Js

2.1 créer un fichier js et l'importer

Nous avons vu dans le cours 1 step0 : nous avons deux fichiers: le js et l'html



- Le fichier js est très simple il fait juste un console.log
=> cela va log quelque chose dans la console et je vais pouvoir voir que le fichier js est bien exécuté.
- le html contient une balise script qui permet d'importer ce js (pas besoin de la suite du fichier pour le moment)

2.1 créer un fichier js et l'importer

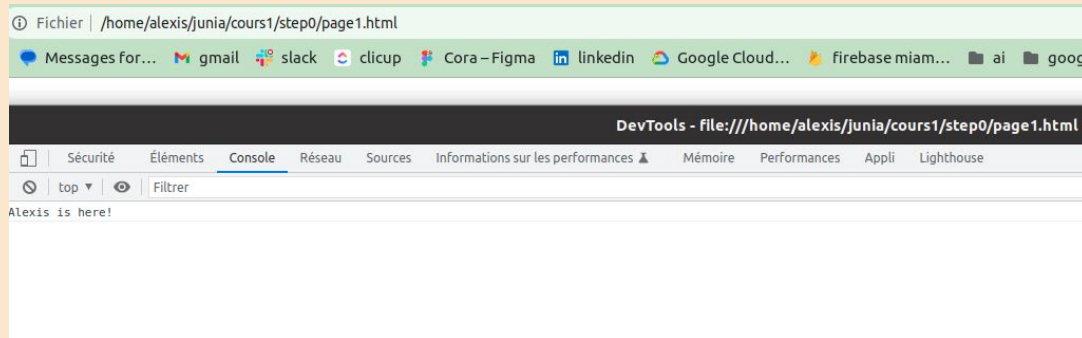
Aide pour réussir : reprendre le cours1 step0. Faire un fichier js très simple avec un log. Regarder le html en particulier la balise script. Adapter le chemin de la balise script. La mettre dans mon propre html

Comment valider ?

1. J'ouvre le html dans mon navigateur
2. j'ouvre la console web
3. je recharge la page
4. je regarde si j'ai bien mon log

Voici un screen ou on voit

- la barre de mon navigateur avec l'adresse de mon fichier
- la console de mon navigateur (titre DevTools), je suis dans la rubrique Console
- Mon log : Alexis is Here!



2.2 les premières interactions : créer une fonction et l'appeler sur tous les boutons

Nous avons vu dans le cours1 step1 comment créer une fonction js :

```
console.log("Alexis is here!")

function changeToBlue() {
  console.log('blue')
}
...
```

Nous avons vu comment appeler ces fonctions sur nos boutons

```
<button onclick="changeToBlue()">Change to blue</button>
<button onclick="changeToRed()">Change to red</button>
<button onclick="changeToYellow()">Change to yellow</button>
```

2.2 les premières interactions : créer une fonction et l'appeler sur tous les boutons

Aide pour réussir : reprendre le cours1 step2.

Créer une fonction unique qui fait juste un `console.log` et utiliser cette même fonction sur tous nos boutons

Comment valider ?

1. J'ouvre le html dans mon navigateur
2. j'ouvre la console web
3. Je clic sur mes boutons
4. je regarde dans la console que j'ai bien mon log

2.3 les premières interactions : De quelles actions ai-je besoin ?

Pour le moment nous avons une fonction unique. Le but ici est d'en avoir plusieurs.

Nous avons vu dans le cours 1 dans un premier temps chaque bouton avait sa propre action. Puis nous sommes passés à une action unique avec un paramètre

Avant :

```
function changeToBlue() {  
  console.log('blue')  
}  
  
function changeToRed() {  
  console.log('red')  
}  
  
function changeToYellow() {  
  console.log('yello')  
}
```

```
<button onclick="changeToBlue()">Change to blue</button>  
<button onclick="changeToRed()">Change to red</button>  
<button onclick="changeToYellow()">Change to  
yellow</button>
```


Après :

```
function changeColor(color) {  
  document.getElementById("change_it").style.backgroundColor = color;  
}
```

```
<button onclick="changeColor('#330aec')">Change to blue</button>  
<button onclick="changeColor('rgb(250, 0, 0)')">Change to red</button>  
<button onclick="changeColor('#c7f014')">Change to yellow</button>
```

Remplacer la fonction unique par plusieurs fonctions :

1. Quelles sont les actions similaires qui vont utiliser la même fonction avec un paramètre ?
2. Quelles sont les actions spécifiques qui ont besoin d'une autre fonction.

=> faire donc x fonction qui ne font qu'un `console.log` différent et les appeler correctement dans le html.

Aide pour réussir : reprendre le cours1 step3 pour avoir la fonction avec paramètre et l'appel.

Comment valider ?

- Commencez par bien réfléchir et essayez de trouver la solution vous même
- regardez les slides suivantes

Pour rappel :

- les 9 chiffres et la virgule/le point
- les opérations +, -, *, /
- les parenthèses
- un bouton pour effacer la dernière opération
- une zone où apparaîtra ce que je tape
- une zone avec le résultat

2.3.1 les premières implémentations - modification du dom

Nous allons commencer par implémenter une fonction d'écriture:

Quand je tape sur les touches numériques, la virgule, les opération ou parenthèses je veux simplement que le texte correspondant à la touche s'ajoute au texte prévu pour voir ce que je tape.

je passe de



à



2.3.1 les premières implémentations - modification du dom

Pour réaliser ceci:

1. Je dois récupérer un élément du dom
2. Je dois appliquer une opération (ajout d'un caractère)
3. je dois modifier un élément du dom

Dans le cours nous avons vu:

2.3.1 les premières implémentations - modification du dom

Aide pour réussir :

Comment valider ?

2.3.2 les premières implémentations - modification du dom

Nous allons maintenant implémenter la fonction de correction. Le but est similaire à la fonction précédente mais au lieu d'ajouter un caractère elle en retire un

2.3.2 les premières implémentations - modification du dom

Aide pour réussir :

Comment valider ?

2.3.3 les premières implémentations - modification du dom

Nous allons maintenant implémenter la fonction de calcul.

Elle a une partie similaire à celles d'avant :

1. Je dois récupérer un élément du dom
2. Je dois appliquer une opération
3. je dois modifier un élément du dom

Regardez la fonction 'eval' et essayez dans une console web

```
> eval('1+2')
```

```
< 3
```

```
> eval('1++2')
```

```
✖ ▶ Uncaught SyntaxError: Invalid left-hand side expression in postfix operation  
   at <anonymous>:1:1
```

```
> |
```


2.3.3 les premières implémentations - modification du dom

Aide pour réussir :

Comment valider ?



Vous avez votre première calculatrice !

Partie 3 : C'est la classe

Partie 4 : Bonus : le retour arrière

Partie 6 : serveur

Partie 7 : calculatrice to server

A propos de Miam

Miam réinvente et simplifie les courses alimentaires.



Qui sommes-nous ? Une startup tech lilloise, qui développe depuis 2019 des solutions d'intelligence artificielle à destination de la grande distribution.

Plus de détails sur <https://miam.tech>