



5IPRO

Principes algorithmiques et programmation

Benjamin Delbar



Cours 4

Feedback & Correction interro

Les différentes familles de langages

Javascript - Rapide histoire

Variables

Types de variables

Opérations arithmétiques

Conditions

Exercices



Feedback & Correction

+ de 75% de réussite!

Interro de rattrapage semaine prochaine

Meilleur des 2 résultats



Les différentes familles de langages de programmation

Par rapport à leur utilisation par le système

- Langage compilé *C, C++, ...*
- Langage interprété *Java, Javascript, ...*

Par rapport au degré d'abstraction

- Langage bas niveau *C*
- Langage haut niveau *Javascript*

Par rapport à leur paradigme

- Langage procédural *BASIC, C, ...*
- Langage orienté objet *Java, C#, ...*
- Langage multi paradigme *Python, Javascript, ...*



Javascript - Rapide histoire

Créé par Brendan Eich en **1995**, en 10 jours

Javascript car au départ pensé pour aller avec du Java

Créé pour aller avec le navigateur Netscape

Adopté par les autres navigateurs

Pas vraiment pris au sérieux

Boosté par Google en 2008 avec Chrome V8

NodeJS en 2009



Javascript - Rapide histoire

Langage interprété

Haut niveau

Prise en charge de l'orienté objet

Forte popularité

Inévitable en web, desktop app, apps

Javascript - Les variables

<https://playcode.io/new/>

Un programme utilise des variables pour stocker nos données

Variable = une boîte qui va garder notre donnée

Donner un nom cohérent



Javascript - Les variables

<https://playcode.io/new/>

Variable = notre boîte

Valeur = contenu de notre boîte

Convention

Nommez vos variables correctement, **en anglais**, et en suivant une logique similaire

Le camelCase : name, age, dateOfBirth, hasChildren, ...

Le _ : firstname, age, date_of_birth, has_children, ...





Javascript - Les variables

Les **erreurs** à ne pas commettre

Le tiret : date-of-birth

Les espaces : date of birth

Les noms peu descriptif : dob, x, y, z

Maj au début : DateOfBirth



Javascript - Les variables

Déclarer une variable en javascript

```
let age = 39;
```

```
let name = "Jean";
```

```
let marriedSince = 1992;
```

```
let group; //undefined
```

Lors de vos recherches

On peut aussi déclarer une variable avec le mot clef var au lieu de let, ne vous attardez pas la dessus pour le moment



Javascript - Les variables

Modifier la valeur d'une variable en JS

```
let age = 39;
```

```
age = 43; //age vaut 43
```

```
age = "Bonjour"; //age vaut "Bonjour"
```



Javascript - Les variables

Exercice

- 1) `revenuMensuel`
- 2) `age`
- 3) `Height`
- 4) `has_pets`
- 5) `numberOfSisters`
- 6) `s`
- 7) `weightOfThePersonImCurrentlyAskingTo`
- 8) `woftpimcat`
- 9) `number_ofStudents`
- 10) `placeOfBirth`

Exercice

Détectez les variables incorrectes / ne respectant pas la convention

Pour les variables incorrectes, trouvez un remplacement



Javascript - Les variables

Exercice

Je m'appelle Paul, j'ai 2 chats et 3 chiens. J'habite à Wavre mais je suis né à Braine-l'Alleud. J'ai 25 ans et je n'ai pas d'enfants.

Avec ces informations déclarez les variables nécessaires

Pas besoin d'y attribuer une valeur pour le moment



Javascript - Les variables

Les constantes

Ne peuvent être modifiée

Se déclare avec le mot clef **const**

```
const PI = 3.14;
```

Convention

On nomme nos constantes tout en majuscule, cela nous permet de les détecter rapidement



Javascript - Les types de variables

En javascript 3 types **primitifs**

number 455, 0.3, -9

string 'hello', "bonjour", "je m'appelle Benjamin", "599"

boolean true, false



Javascript - Type Number

number

Peut être positif ou négatif

Nombres entiers => **integers** => **int**

Nombres à virgules => **floating point** => **float**

Attention

Les floats ont un comportement un peu bizarre, lors de calcul précis on les évitera. Au lieu de calculer une taille en mètre (let height = 1.80) on va la calculer en cm (let height = 180)



Javascript - Type String

string

Chaîne de caractère, on encapsule nos strings dans des quotes “ ou des single quotes ‘

```
let name = “Benjamin”;
```

Attention

Rien ne nous empêche d’écrire un nombre dans une chaîne de caractère
let pseudo = “57”; let introduction = “Bonjour, j’ai 2 enfants”, etc



Javascript - Type Boolean

boolean

Vrai / Faux

```
let has_children = false;  
let isAlive = true;
```

Convention

On essaye **au maximum** d'utiliser des noms de variables en hasXYZ et isXYZ pour détecter rapidement nos booléens



Javascript - Types

Exercice

Je m'appelle Paul, j'ai 2 chats et 3 chiens. J'habite à Wavre mais je suis né à Braine-l'Alleud. J'ai 25 ans et je n'ai pas d'enfants.

Avec ces informations déclarez les variables nécessaires et attribuez la valeur



Javascript - Types

Exercice

```
1) let hasChildren = 'Non';
2) let myName = Benjamin;
3) let IsAlive = true;
4) let amount = 500;
5) let numberOfStudents = '20';
6) let address = "1348, Louvain-la-Neuve";
7) let has_grades = "false";
8) let height = 1.80;
9) let average-Grade = "La moyenne est de 12";
10) const std = false;
```

Exercice

Détectez :

Le type de la variable et si c'est un bon choix

Si le nom est correct et si la valeur est correcte

Corrigez en fonction



Javascript - Opérations arithmétiques

Number

Addition (+) Soustraction (-) Multiplication (*) Division (/) Modulo (%)

Incrémentation / Décrémentation autorisée (x++, ++x, x--, --x)

Simplification autorisée ($x *= 2 \Rightarrow x = x * 2$)

String

Concaténation (+)

Boolean

Opérations arithmétiques sur booléen = comportement spécial

true est interprété comme valant 1 et false comme 0



Javascript - Conditions

Opérateurs relationnels

==	comparaison de valeur
===	comparaison de valeur et de type
!=	différence de valeur
!==	différence de valeur et de type
<= < > >=	strict. plus petit / plus petit ou égal / strict. plus grand / plus grand ou égal



Javascript - Conditions

If ...

```
if (condition) {  
    // écrire le code qui s'exécute ici si la condition est remplie  
}
```

If ... Else

```
if (condition) {  
    // écrire le code qui s'exécute ici si la condition est remplie  
} else {  
    // écrire ce qui se passe si la condition n'est pas remplie  
}
```



Javascript - Conditions

If ... Else If .. Else

```
if (condition) {  
    // écrire le code qui s'exécute si la condition est remplie  
} else if (autre condition) {  
    // écrire ce qui se passe si c'est cette condition qui est remplie  
} else {  
    // dans les autres cas  
}
```

Attention

Il n'est pas obligatoire d'avoir un else, on peut avoir une condition if .. else if .. uniquement, on peut aussi avoir plusieurs else if



Javascript - Conditions

```
let age = 10;
```

```
if (age < 18) {  
    // Le code ici s'exécute  
} else if (age == 10) {  
    // la condition est remplie mais le code a déjà trouvé une condition valable  
} else if (age > 8) {  
    // la condition est remplie mais le code a déjà trouvé une condition valable  
} else {  
    //dans les autres cas  
}
```

Attention

On rentre dans la **première** condition valable et on n'exécute pas le code des autres



Javascript - Conditions

```
let age = 59;  
  
if (age) {  
    // do something  
}
```

équivalent à faire “est-ce que age existe et n’est pas égal à 0?”



Javascript - Conditions

```
let hasChildren = true;

if (hasChildren) {
    // do something si il a des enfants
}

-----

let hasChildren = false;

if (!hasChildren) {
    // do something si il n'a pas d'enfants
}
```



Javascript - Afficher une valeur

Dans un premier temps :

```
console.log(maVariable);
```



Javascript - Exercices

Réalisez le code nécessaire pour effectuer les 2 flow charts du cours 1

Réalisez le code nécessaire pour effectuer le calcul de l'IMC du cours 2

Réalisez le code nécessaire pour effectuer les questions 2 et 5 de l'interrogation