

Cours 8

Fonctions avec paramètres

Intro. à la programmation - Aut. 2021

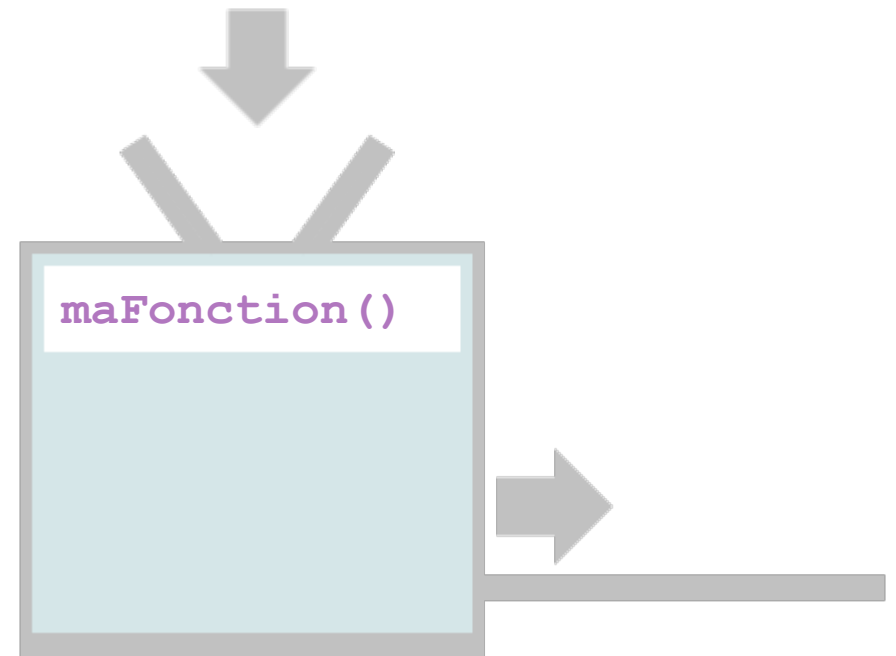


- ❖ Fonctions sans paramètres
- ❖ Fonctions avec paramètres



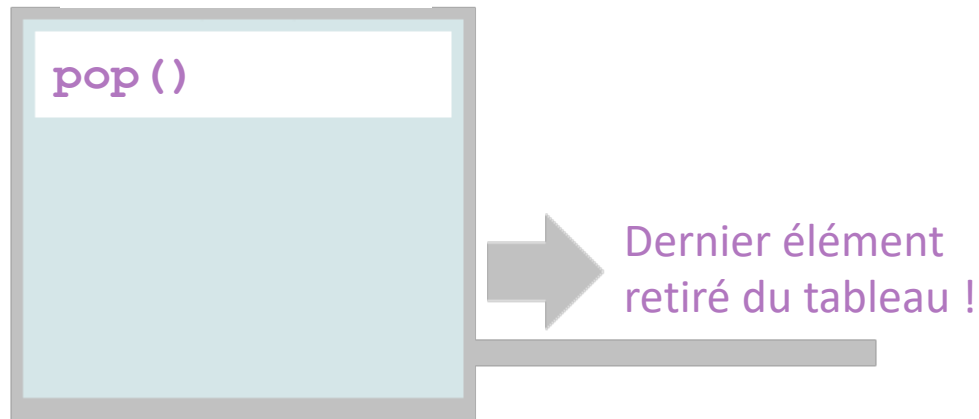
❖ Une **fonction** est un peu comme une **machine**

- ◆ On démarre la machine
 - On **appelle** la fonction
- ◆ La machine fait quelque chose
 - La **fonction** fait quelque chose

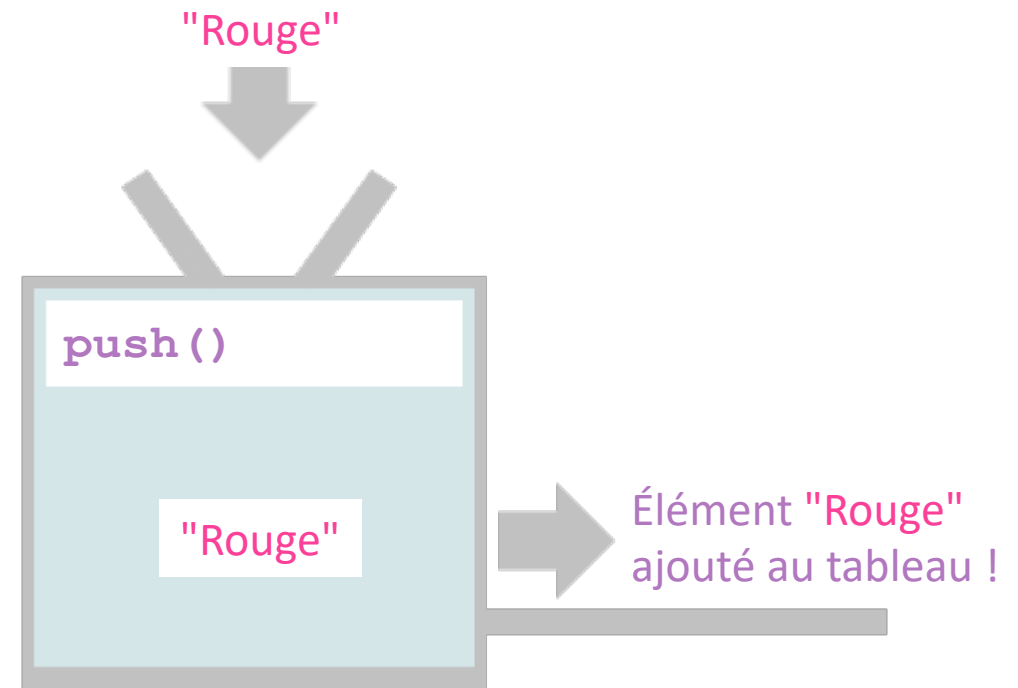




- ❖ Certaines fonctions n'ont pas besoin de *données supplémentaires* pour fonctionner, alors que d'autres oui !



Pas besoin de fournir de donnée pour **pop()** !



Il faut préciser à **push(...)** ce qu'on veut ajouter au tableau ! (Ici ... **"Rouge"**)



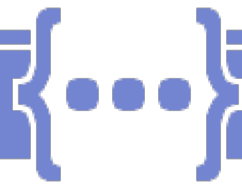
❖ **texteRouge ()** est un exemple de fonction sans paramètres.

- ◆ On peut reconnaître une fonction sans paramètres :
 - Dans sa déclaration, les **parenthèses sont vides**.



```
function texteRouge() {  
    this.style.color = "red";  
}
```

Certains des exemples qui vont suivre ne sembleront pas très utiles, mais nous abordons des exemples qui sont simples avant tout, afin de viser une meilleure compréhension.



❖ Certaines fonctions ont besoin de données (de « **paramètres** ») pour être appelées.

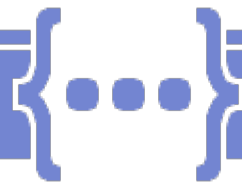
◆ À quoi ressemble une fonction qui nécessite un ou plusieurs paramètres ?

○ Syntaxe :

maFonction a besoin de **2 paramètres** pour être appelée.

```
function maFonction(param1, param2)  
{  
    // Instructions  
}
```

On peut donner n'importe quel **nom** aux **paramètres**. L'important est qu'un nom représente bien l'utilité du paramètre.



❖ Fonctions avec un seul paramètre

- ◆ Exemple 1 : La fonction **colorerTexte** a besoin d'un seul paramètre. Ce paramètre doit être une chaîne de caractères qui représente une couleur qui pourra être reconnue par CSS.

La déclaration de la fonction :

```
function colorerTexte(couleur){  
    document.getElementById( elementId: "message").style.color = couleur;  
}
```

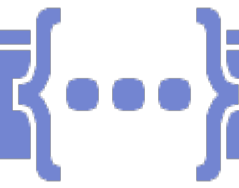
L'appel de la fonction : (On choisit la couleur !)

```
colorerTexte( couleur: "blue");
```



```
function colorerTexte(couleur){  
    document.getElementById( elementId: "message").style.color = couleur;  
}
```

Dans ce cas-ci, **couleur** vaut "blue" pour cet appel de la fonction.



❖ Fonctions avec un seul paramètre

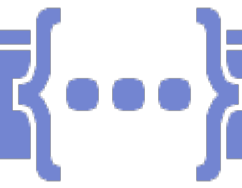
- ◆ Exemple 2 : La fonction **majeurOuMineur** a besoin d'un paramètre de type **nombre**.

Déclaration de la fonction

```
function majeurOuMineur(age){  
    if(age >= 18){  
        gMajeur = true;  
    }  
    else{  
        gMajeur = false;  
    }  
}
```

Exemples d'appels

```
majeurOuMineur( age: 8);  
// gMajeur vaut maintenant false  
  
majeurOuMineur( age: 24);  
// gMajeur vaut maintenant true
```

❖ Fonctions avec un seul paramètre

- ◆ Exemple 3 : **texteImportant** a besoin d'un paramètre de type booléen.

Déclaration de la fonction

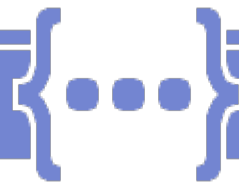
```
function texteImportant(plusGrand){  
    document.getElementById( elementId: "message").style.color = "red";  
    if(plusGrand == true){  
        document.getElementById( elementId: "message").fontSize = "40px";  
    }  
}
```

Exemple d'appel

```
texteImportant( plusGrand: false);  
// Le texte de #message devient rouge
```

Exemple d'appel

```
texteImportant( plusGrand: true);  
// Le texte de #message devient rouge et 40px en hauteur
```



❖ Fonctions avec plusieurs paramètres

◆ Exemple 4 : **colorerElements** a besoin de 2 paramètres :

- Une chaîne de caractères qui représente l'**id** de l'élément qu'on veut modifier.
- La **couleur** qu'on veut donner au texte de l'élément.

Déclaration de la fonction

```
function colorerElement(id, couleur){  
    document.getElementById(id).style.color = couleur;  
}
```

Appel de la fonction

```
colorerElement(id: "message", couleur: "red");
```

Le texte de l'élément avec l'id **message** va devenir **rouge**.



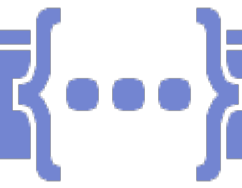
❖ Fonction **sans paramètre** dans une boucle

- ◆ Exemple 5 : Fonction qui retire 10 points de vie à un personnage.
 - On peut se servir d'une boucle pour appeler une fonction plusieurs fois d'affilée.

```
let gPointsDeVie = 100;  
  
function attaquer() {  
    gPointsDeVie -= 10;  
}
```

```
for(let index = 0; index < 4; index++){  
    attaquer(); // Sera appelée 4 fois  
}
```

gPointsDeVie vaudra **60** après la boucle.



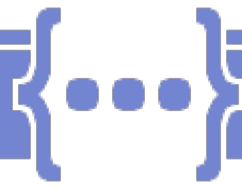
❖ Fonction avec paramètre dans une boucle

- ◆ Exemple 6 : Fonction qui retire x points de vie à un personnage.
 - Cette fois on choisit la force de l'attaque !

```
let gPointsDeVie = 100;  
  
function attaquerMieux(force){  
    gPointsDeVie -= force;  
}
```

```
for(let index = 0; index < 3; index++){  
    attaquerMieux(force: 25); // Appelée 3 fois  
}
```

gPointsDeVie vaudra **25** après la boucle.



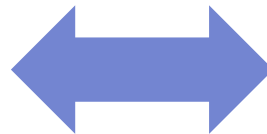
❖ On peut se servir de variables comme **paramètres**.

◆ Exemple 7 : Ci-dessous, une fonction similaire à tout à l'heure...

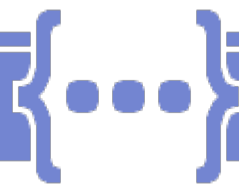
```
function colorerTexteAvecVariable(couleur){  
    document.getElementById( elementId: "message").style.color = couleur;  
}
```

◆ Toutefois, on peut très bien glisser une **variable** comme paramètre, plutôt que directement mettre une **chaîne de caractères**.

```
let maCouleur = "gold";  
colorerTexteAvecVariable(maCouleur);
```



```
colorerTexteAvecVariable( couleur: "gold");
```



❖ On peut se servir de variables ET valeurs comme **paramètres** :

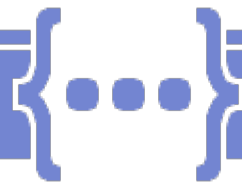
- ◆ Exemple 8 : Encore un exemple similaire, mais nous allons utiliser à la fois une variable et une valeur comme paramètres.

```
function colorerElementAvecVariableEtValeur(id, couleur){  
    document.getElementById(id).style.color = couleur;  
}
```

```
let monId = "message";  
  
colorerElementAvecVariableEtValeur(id, couleur: "gold");
```



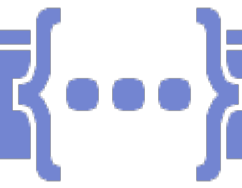
```
colorerElementAvecVariableEtValeur(id: "message", couleur: "gold");
```



- ❖ Exemple 9 : Fonction qui a besoin d'un **tableau d'ids** comme **paramètre** et change la couleur de la bordure en noir pour tous les éléments.

```
function bordureNoireSurPlusieursElements(tableauIds){  
    for(let index = 0; index < tableauIds.length; index++){  
        document.getElementById(tableauIds[index]).style.borderColor = "black";  
    }  
}
```

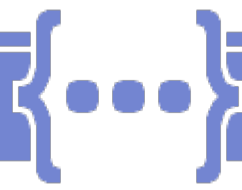
```
let tableauIdsImages = ["imagePitou", "imageMinou", "imageHibou];  
  
bordureNoireSurPlusieursElements(tableauIdsImages);
```

- ❖ Exemple 10 : Fonction qui a besoin d'un **tableau d'ids** et d'une **couleur** (chaîne de caractères) comme **paramètres** et change la couleur de la bordure pour tous les éléments.

```
function colorerBordureSurPlusieursElements(tableauIds, couleur){  
    for(let index = 0; index < tableauIds.length; index++){  
        document.getElementById(tableauIds[index]).style.borderColor = couleur;  
    }  
}
```

```
let tableauIdsImages = ["imagePitou", "imageMinou", "imageHibou"];  
  
colorerBordureSurPlusieursElements(tableauIdsImages, couleur: "gold");
```

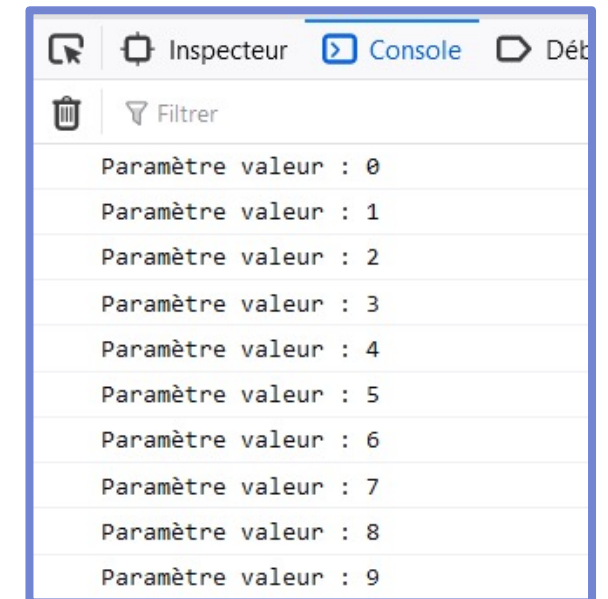



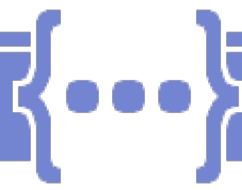
❖ Fonction avec **paramètre** dans une boucle

- ◆ Exemple 11 : Fonction qui imprime un message dans la **console** et qui prend un **paramètre**.

```
function messageConsole(valeur){  
    console.log("Paramètre valeur : " + valeur);  
}
```

```
for(let index = 0; index < 10; index++){  
    messageConsole(index);  
}
```





❖ Fonction avec **paramètre** dans une boucle

- ◆ Exemple 12 : Fonction qui applique un **rabais** sur un des prix du tableau `gTableauxPrix`. On choisit l'**index** (le prix réduit) et la **valeur** du rabais.

```
let gTableauPrix = [20, 15, 10, 18, 22];  
  
function rabaisSurPrix(indexDuPrix, rabais){  
    gTableauPrix[indexDuPrix] -= rabais;  
}
```

```
for(let index = 0; index < gTableauPrix.length; index++){  
    rabaisSurPrix(index, rabais: 5);  
}
```

`gTableauPrix` vaudra [15, 10, 5, 13, 17] après la boucle.