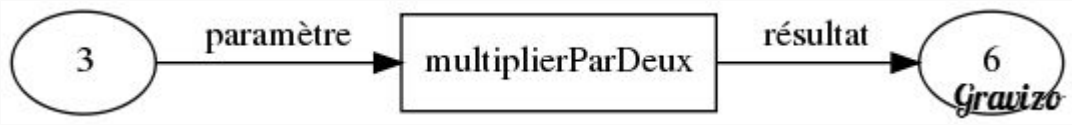
Javascript - Fonctions

Définition et appel de fonction

Par exemple:




```
let resultat = multiplierParDeux(3); // => le paramètre nombre vaut 3 => la variable resultat vaudra 6
```

Comme pour une variable, l'appel à une fonction sera remplacé la valeur qu'elle renvoie, au moment de l'exécution du programme. Contrairement aux variables, cette valeur dépendra de la valeur des paramètres passés à la fonction.

Ainsi, il est possible de passer le résultat de l'appel d'une fonction en paramètre d'une fonction.

Pratique: Définition et appel de fonction

Dans cette partie de mise en pratique, nous allons définir ensemble plusieurs fonctions, et les tester en les appelant.

Développer:

- une fonction `diviserParDeux` qui retourne la moitié de la valeur passée en paramètre. Tests:
  - `console.log( diviserParDeux(2) === 1 );`
  - `console.log( diviserParDeux(4) === 2 );`

- une fonction `somme` qui retourne la somme des deux paramètres qui lui seront passés. Tests:
  - `console.log( somme(1, 1) === 2 );`
  - `console.log( somme(1, 2) === 3 );`
  - `console.log( somme(2, 1) === 3 );`

Pratique: Définition et appel de fonction

- une fonction `signe` qui retourne la chaîne de caractères `positif`, `négatif` ou `nul`, selon le signe de la valeur passée en paramètre. Tests:
  - `signe(-1) === 'négatif';`
  - `signe(0) === 'nul';`
  - `signe(2) === 'positif';`

- une fonction `factorielle` qui retourne le produit de tous les entiers consécutifs entre 1 et l'entier passé en paramètre (compris). Exemple: `factorielle(3)` retourne le résultat de `1 * 2 * 3`, soit `6`. Tests:
  - `factorielle(1) === 1;`
  - `factorielle(3) === 6;`
  - `factorielle(4) === 24;`

Exercice: Jeu Chi Fou Mi

Fonctionnement du jeu à implémenter

À chaque manche, l'ordinateur et le joueur choisissent chacun un élément parmi `pierre`, `feuille` ou `ciseaux`.

Un point est donné à celui qui a choisi l'élément le plus fort, sachant que:

- `ciseaux` > `feuille` *(les ciseaux coupent la feuille)*
- `pierre` > `ciseaux` *(la pierre casse les ciseaux)*
- `feuille` > `pierre` *(la feuille enveloppe la pierre)*

Si l'ordinateur et le joueur ont choisi le même élément, aucun d'eux n'emporte de point.

Exemple de déroulement d'une manche

- l'ordinateur choisit secrètement `pierre` (parmi les trois valeurs d'éléments possibles);
- le joueur est invité à saisir son choix => il tape `feuille`;
- l'ordinateur affiche `feuille` car c'est l'élément qui l'emporte (la feuille enveloppe la pierre).

Exercice: Jeu Chi Fou Mi - Phase 1 - Implémentation d'une manche

Pour implémenter le code d'une manche, nous allons:

- définir une fonction `comparer(choix1, choix2)` qui renvoie le nom de l'élément gagnant, entre les deux passés en paramètres;
- appeler cette fonction, en passant les choix de l'ordinateur et du joueur en paramètres, afin de savoir lequel des deux a remporté la manche.

Pour vous aider, le site codecademy propose un guide interactif: Créez un "Pierre, feuille, ciseaux".

Si vous ne souhaitez pas utiliser ce site, voici une proposition d'étapes à suivre:

1. Dessiner l'arbre de décision d'une manche: nom de l'élément gagnant en fonction de deux éléments choisis;
2. Transformer l'arbre de décision en conditions if imbriquées, en fonction de la valeur de deux variables: `choix1` et `choix2`;
3. Chaque condition de dernier niveau va afficher dans la console le nom de l'élément qui remporte la manche;
4. Transférer ces conditions dans la définition d'une fonction `comparer(choix1, choix2)` qui retourne le nom de l'élément gagnant à l'aide de `return` (au lieu de l'afficher dans la console), parmi les deux passés en paramètres; Si les deux sont identiques, retourner `égalité`.
5. Tester cette fonction en lui passant chaque combinaison possible de valeurs du jeu en paramètres;
6. En dehors de la définition de la fonction, créer les variables `choixOrdi` et `choixUtilisateur`;
7. Faire en sorte que `choixOrdi` ait pour valeur un des trois éléments, choisi de manière aléatoire, et que `choixUtilisateur` soit saisi par l'utilisateur à l'aide de `prompt()`;
8. Appeler la fonction `comparer()`, puis afficher dans la console la valeur de son résultat (l'élément qui remporte la manche), à partir des choix de l'ordinateur et du joueur.

Exercice: Jeu Chi Fou Mi - Phase 2 - Partie en 3 manches, et scores

Après avoir implémenté une manche à l'aide de la fonction `comparer()`, faites en sorte que le joueur puisse jouer 3 manches d'affilée et que le score final du joueur et de l'ordinateur soient affichés dans la console en fin de partie.

Pour cela:

1. Créer les variables `scoreOrdi` et `scoreJoueur`;
2. Après l'affichage du résultat de l'appel à `comparer()` dans la console, incrémenter une de ces variables, en fonction de qui a remporté la manche;
3. Mettre le code correspondant à une manche dans une boucle `for`, de manière à ce qu'il s'exécute `3` fois d'affilée;
4. En fin de partie, afficher qui a remporté la partie: `'ordi'`, `'joueur'` ou `'aucun'`, en fonction des scores.