

Hiver 2019Professeur : Amor Amami

Pratique : les fonctions en R

1. Créez une fonction qui prend un nom en tant qu'argument de chaîne et affiche "Hello name"

```
hello_you <- function(name){  
  print(paste('Hello',name))  
}
```

```
hello_you('Sam')
```

2. Créez une fonction qui prend un nom en tant qu'argument de chaîne et **retourne** une chaîne de la forme - "Hello name"

```
hello_you2 <- function(name){  
  return(paste('Hello',name))  
}
```

```
print(hello_you2('Sam'))
```

3. Créez une fonction qui retournera le produit de deux entiers.

```
prod <- function(num1,num2){
```

```
    return(num1*num2)
}
```

```
prod(3,4)
```

4. Créez une fonction qui accepte deux arguments, un entier et un vecteur d'entiers. Il renvoie VRAI si l'entier est présent dans le vecteur, sinon FALSE. Assurez-vous de porter une attention particulière à la position de la ligne de retour (FAUX) dans votre fonction!

```
num_check <- function(num,v){
  for (item in v){
    if (item == num){
      return(TRUE)
    }
  }
  return(FALSE)
}
```

```
num_check(2,c(1,2,3))
```

```
num_check(2,c(1,4,5))
```

5. Créez une fonction qui accepte deux arguments, un entier et un vecteur d'entiers. Il renvoie le nombre d'occurrences de l'entier dans le vecteur d'entrée.

```
num_count <- function(num,v){
  count = 0
  for (x in v){
    if (x == num){
```

```

        count = count + 1
    }
}
return(count)
}

```

```
num_count(2,c(1,1,2,2,3,3))
```

```
num_count(1,c(1,1,2,2,3,1,4,5,5,2,2,1,3))
```

6. Nous voulons expédier des barres d'aluminium. Nous allons créer une fonction qui accepte un entier représentant les kilogrammes d'aluminium demandés pour le colis à expédier. Pour répondre à ces demandes, nous disposons de petites barres (1 kilogramme chacune) et de grandes barres (5 kilogrammes chacune). Renvoie le moins de barres nécessaires.

Par exemple, une charge de 6 kg nécessite un minimum de deux barres (1 barre de 5 kg et 1 barre de 1 kg). Une charge de 17 kg nécessite un minimum de 5 barres (3 barres de 5 kg et 2 barres de 1 kg).

```

bar_count <- function(pack){
  amount_of_ones = pack %% 5
  amount_of_fives = (pack - amount_of_ones)/5
  return(amount_of_ones+amount_of_fives)
}

```

```
bar_count(6)
```

```
bar_count(17)
```

7. Créez une fonction qui accepte 3 valeurs entières et renvoie leur somme. Cependant, si une valeur entière est divisible par 3, elle ne

compte pas dans la somme. Renvoie zéro si tous les nombres sont divisibles par 3. Astuce: vous pouvez utiliser la fonction `append()`.

```
summer <- function(a, b, c){  
  out <- c(0)  
  if (a %% 3 != 0){  
    out <- append(a,out)  
  }  
  if (b %% 3 != 0){  
    out <- append(b,out)  
  }  
  if (c %% 3 != 0){  
    out <- append(c,out)  
  }  
  return(sum(out))  
}
```

```
summer(7,2,3)
```

```
summer(3,6,9)
```

```
summer(9,11,12)
```

8.

Créez une fonction qui retournera TRUE si un entier en entrée est premier. Sinon, retourne FALSE.

Beaucoup de façons de résoudre celui-ci! Nous ne nous préoccupons pas trop des cas extrêmes tels que les nombres négatifs, mais n'hésitez pas à ajouter des déclarations si pour vérifier celles-ci!

Un nombre **a** est divisible par un nombre **b** si le résultat de la division **a / b** est égal au résultat de la division entière **a % /% b**. Tout nombre entier **pn** peut être divisé par au moins deux nombres: **1** et **pn**. Les

nombre premiers sont ceux qui ne peuvent être divisés que par ces deux. Sortir le code:

- **pn / 1: pn** sont les résultats des divisions par 1, 2, ..., pn
- **pn%/% 1: pn** sont les résultats des divisions entières par 1, 2, ..., pn
- **sum (pn / 1: pn == pn%/% 1: pn)** sont combien de ceux-ci sont égaux, c'est-à-dire le nombre de diviseurs entiers de pn. Si ce nombre est 2, vous avez un nombre premier.

```
prime_check <- function(num) {  
  if (num == 2) {  
    return(TRUE)  
  } else if (any(num %% 2:(num-1) == 0)) {  
    return(FALSE)  
  } else {  
    return(TRUE)  
  }  
}
```

```
# Alternatively:  
prime_check <- function(num){  
  # Could put more checks for negative numbers etc...  
  if (num == 2) {  
    return(TRUE)  
  }  
  for (x in 2:(num-1)){  
  
    if ((num%%x) == 0){  
      return(FALSE)  
    }  
  }  
  return(TRUE)  
}
```

```
prime_check(2)
```

```
prime_check(4)
```

```
prime_check(131)
```