

# Atelier PYTHON Day One

Winter

21 mars 2015

# Introduction :

Mail : [winter@lebib.org](mailto:winter@lebib.org)

Le vendredi de 20h à 21h30

## Un peu d'histoire :



FIGURE : Guido Van Rossum

1990-1991 : Apparition

1999 : v1.5

2000 : v2.0

2010 : v2.7

2008 : v3.0

2014 : v3.4

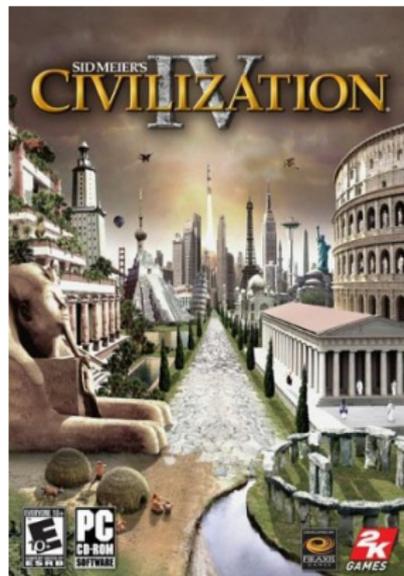
# Utilisation :



FIGURE : Blender



FIGURE : Scipy



# Venimeux ?



Pour un bon programme... il faut de bons outils :



FIGURE : Terminal

# Une question de type :

1. Nombres
2. Chaînes de caractères
3. Booléens

# Théorie des nombres :

1. Entiers
2. Flottants

# Int :

Taille limitée :

- ▶  $-2^{31}$  et  $2^{31} - 1$  : 32-bits
- ▶  $-2^{63}$  et  $2^{63} - 1$  : 64-bits

# Float :

Les nombres flottants sont stockés suivant l'écriture scientifique.

$$\textit{mantisse} \cdot 10^{\textit{exp}}$$

# Float :

Les nombres flottants sont stockés suivant l'écriture scientifique.

$$\textit{mantisse} \cdot 10^{\textit{exp}}$$

## Example - 2015

Mantisse : 2.015

Exposant : 3

Ecriture :  $2.015 \cdot 10^3$

# Écriture :

Comment écrire nos nombres dans Python ?

## int

```
>>> 1
>>> -1337
>>> +42
```

## float

```
>>> 1.0
>>> 1
>>> -0.2
>>> .2
>>> 1.337e42
>>> 1.3337e-42
```

# Strings :

Il existe plusieurs possibilités pour écrire des chaînes de caractères en Python.

```
>>> 'Hello, World'  
>>> "Hello, World!"  
>>> """Hello, World!"""
```

# Strings :

Il existe plusieurs possibilités pour écrire des chaînes de caractères en Python.

```
>>> 'Hello, World'  
>>> "Hello, World!"  
>>> """Hello, World!"""
```

## Problème :

```
'J'aime les chats.'
```

# Strings :

Il existe plusieurs possibilités pour écrire des chaînes de caractères en Python.

```
>>> 'Hello, World'  
>>> "Hello, World!"  
>>> """Hello, World!"""
```

## Problème :

```
'J'aime les chats.'
```

## Solutions :

```
>>> "J'aime les chats."  
>>> 'J\'aime les chats.'
```

# Booléens :

Deux valeurs possibles : Vrai ou Faux.  
Deux états possibles : 1 ou 0.  
Ils sont écrits `True` et `False` en Python.

# Opérations

1. Opérations de base.
2. Opérations supplémentaires.
3. Opérations booléennes.

# Opérations de base :

# Opérations de base :

## Concaténation I :

```
>>> "J'aime" + " les " + "chats."  
"J'aime les chats."
```

# Opérations de base :

## Concaténation I :

```
>>> "J'aime" + " les " + "chats."  
"J'aime les chats."
```

## Concaténation II :

```
>>> "Miam ! " * 5  
'Miam ! Miam ! Miam ! Miam ! Miam ! '
```

# Opérations de base :

## Concaténation I :

```
>>> "J'aime" + " les " + "chats."  
"J'aime les chats."
```

## Concaténation II :

```
>>> "Miam ! " * 5  
'Miam ! Miam ! Miam ! Miam ! Miam ! '
```

## Opérations élémentaires :

```
>>> 0 + 1:1.  
>>> 0 - 1:-1.  
>>> 2 * 1:2.  
>>> 1 / 2:0.5.
```

# Opérations supplémentaires :

## Opérations supplémentaires :

Division entière :

```
>>> 3 // 2
```

```
1
```

## Opérations supplémentaires :

### Division entière :

```
>>> 3 // 2
```

```
1
```

### Puissance :

```
>>> 2 ** 2
```

```
4
```

## Opérations supplémentaires :

### Division entière :

```
>>> 3 // 2  
1
```

### Puissance :

```
>>> 2 ** 2  
4
```

### Modulo

```
>>> 10 % 3  
1
```

# Opérations booléennes :

## AND

```
>>> False and False -> False
>>> False and True  -> False
>>> True and False  -> False
>>> True and True   -> True
```

## OR

```
>>> False or False -> False
>>> False or True  -> True
>>> True or False  -> True
>>> True or True   -> True
```

## NOT

```
>>> not True
False
>>> not False
True
```

# Noms :

On peut assigner des “noms” à des expressions ou résultats.  
Cependant il faut respecter quelques règles :

## Obligatoire :

- ▶ Ne doit pas commencer par un chiffre.
- ▶ Ne doit pas avoir des caractères tels que les opérateurs ou `.`
- ▶ Ne doit pas correspondre aux keywords.

# Questions ?

Et maintenant c'est à vous de jouer !

# Nombres

## Expressions :

```
2 / 0
```

```
10 // 3
```

```
2 * 3
```

```
2 ** 3
```

```
4 % 1
```

```
4 % 2
```

```
3 % 2
```

# Nombres

## Expressions :

```
2 / 0      ZeroDivisionError : division by zero
```

```
10 // 3
```

```
2 * 3
```

```
2 ** 3
```

```
4 % 1
```

```
4 % 2
```

```
3 % 2
```

# Nombres

## Expressions :

```
2 / 0      ZeroDivisionError : division by zero
10 // 3    3
2 * 3
2 ** 3
4 % 1
4 % 2
3 % 2
```

# Nombres

## Expressions :

```
2 / 0      ZeroDivisionError : division by zero
10 // 3    3
2 * 3      6
2 ** 3
4 % 1
4 % 2
3 % 2
```

# Nombres

## Expressions :

```
2 / 0      ZeroDivisionError : division by zero
10 // 3    3
2 * 3      6
2 ** 3     8
4 % 1      4
4 % 2      0
3 % 2      1
```

# Nombres

## Expressions :

```
2 / 0      ZeroDivisionError : division by zero
10 // 3    3
2 * 3      6
2 ** 3     8
4 % 1      0
4 % 2
3 % 2
```

# Nombres

## Expressions :

<code>2 / 0</code>	<code>ZeroDivisionError : division by zero</code>
<code>10 // 3</code>	<code>3</code>
<code>2 * 3</code>	<code>6</code>
<code>2 ** 3</code>	<code>8</code>
<code>4 % 1</code>	<code>0</code>
<code>4 % 2</code>	<code>0</code>
<code>3 % 2</code>	

# Nombres

## Expressions :

<code>2 / 0</code>	<code>ZeroDivisionError : division by zero</code>
<code>10 // 3</code>	<code>3</code>
<code>2 * 3</code>	<code>6</code>
<code>2 ** 3</code>	<code>8</code>
<code>4 % 1</code>	<code>0</code>
<code>4 % 2</code>	<code>0</code>
<code>3 % 2</code>	<code>1</code>

# Chaînes de caractères :

## Expressions :

```
"ah" * 5
```

```
"J'aime "+ "les " + "chats."
```

```
'César : "Alea Jacta Est".'
```

```
"César : "Alea Jacta Est"."
```

# Chaînes de caractères :

## Expressions :

```
"ah" * 5
```

```
'ahahahahah'
```

```
"J'aime "+ "les " + "chats."
```

```
'César : "Alea Jacta Est".'
```

```
"César : "Alea Jacta Est"."
```

# Chaînes de caractères :

## Expressions :

```
"ah" * 5
```

```
"J'aime "+ "les " + "chats."
```

```
'César : "Alea Jacta Est".'
```

```
"César : "Alea Jacta Est"."
```

```
'ahahahahah'
```

```
'J'aime les chats.'
```

# Chaînes de caractères :

## Expressions :

```
"ah" * 5
```

```
"J'aime "+ "les " + "chats."
```

```
'César : "Alea Jacta Est".'
```

```
"César : "Alea Jacta Est"."
```

```
'ahahahahah'
```

```
'J'aime les chats.'
```

```
'César : "Alea Jacta Est".'
```

# Chaînes de caractères :

## Expressions :

```
"ah" * 5
```

```
"J'aime "+ "les " + "chats."
```

```
'César : "Alea Jacta Est".'
```

```
"César : "Alea Jacta Est"."
```

```
'ahahahahah'
```

```
'J'aime les chats.'
```

```
'César : "Alea Jacta Est".'
```

```
SyntaxError : invalid syntax
```