

Les dates en python

Les dates en python

- Pour créer une variable date, il faut importer le module **datetime** (de la même manière qu'on a fait pour le module math).
- Le module **datetime** définit les types suivants :
 - **date** : représente une date; possède 3 paramètres qui représentent successivement : l'année, le mois et le jour
 - **time** : représente un temps; possède des paramètres qui représente l'heure, la minute et la seconde
 - **datetime** : combinaison d'une date et d'un temps; possède des paramètres qui représentent l'année, le mois, le jour, l'heure, la minute et la seconde....

Créer une nouvelle date

- En utilisant la classe **datetime**:
- Syntaxe:

```
import datetime
```

```
Variable=datetime.datetime(annee, mois, jour, heure, minutes, secondes)
```

- Exemple:

```
import datetime
x=datetime.datetime(2023,12,13,5,23,40)
print(x)
```

- Résultat:

```
2023-12-13 05:23:40
```

Créer une nouvelle date

- En utilisant la classe `date` :

- *Syntaxe:*

```
import datetime
```

```
Variable=datetime.date(annee, mois, jour)
```

- *Exemple:*

```
import datetime  
x=datetime.date(2023,5,12)
```

- *Résultat :*

```
2023-12-13
```

Créer un valeur de temps

- En utilisant la classe `time` :
- **Syntaxe:**

```
import datetime
```

```
Variable=datetime.time(heure, minutes, secondes)
```

- **Exemple:**

```
import datetime
x=datetime.time(5,20,51)
print(x)
```

- **Résultat:**

```
05:20:51
```

Obtenir la date et temps actuelle

- Pour obtenir la date système (date d'aujourd'hui et le temps actuel), on utilise la méthode `now()`, disponible uniquement pour la classe `datetime`
- Il existe aussi la méthode `today()`, qui disponible pour la classe `datetime` et la classe `date`

```
import datetime  
  
variable=datetime.datetime.now()
```

```
import datetime  
  
variable=datetime.datetime.today()
```

```
import datetime  
  
variable=datetime.date.today()
```

Obtenir l'année, le mois et le jour d'une certaine date

- Une fois on crée une variable date, on peut obtenir l'année , le mois, le jour.... chacune à part à l'aide des propriétés suivantes :
 - `year` : retourne l'année d'une date donnée
 - `month` : retourne le mois d'une date donnée
 - `day` : retourne le jour d'une date donnée
 - `hour` : retourne l'heure d'un temps donnée (pour datetime)
 - `minute` : retourne la minute d'un temps donnée (pour datetime)
 - `second` : retourne la seconde d'un temps donnée (pour datetime)

Obtenir l'année, le mois et le jour d'une certaine date

- Exemple de code:

```
import datetime
d=datetime.datetime(2003,5,24)
print(d.year)
print(d.month)
print(d.day)
```

- Le résultat:

```
2003
5
24
```


Saisir une date et la convertir en objet date

- Tous ce qui est saisie sur le terminal est une chaine de caractère. Donc il faudra savoir **comment convertir une chaine en type date**.
- D'abord on demande à l'utilisateur de saisir une date avec une certaine forme puis on fait la conversion à l'aide de la fonction **strptime()**

La méthode `strptime`

- La méthode **`strptime()`** est utilisée pour convertir une chaîne en un objet `datetime`.
- Cette méthode accepte deux paramètres, le premier c'est la date fournie, et le deuxième paramètre c'est le **format** de l'objet date souhaité.

Exemple de conversion

- Exemple:

```
x=input("saisir votre date de naissance sous forme jj-mm-aaaa\n")  
  
dd=datetime.datetime.strptime(x,"%d-%m-%Y")  
print(dd)  
print(dd.year)  
print(dd.month)  
print(dd.day)
```

Format de conversion

Format de chaine

- Voici un tableau explicatif :

Directive	Signification	Exemple
%Y	Année sur 4 chiffres	2015,2022,...
%y	Année sur 2 chiffres	15,22,...
%m	Mois en décimal	01, 02, ..., 12
%d	Jour du mois	01, 02, ..., 31
%H	Heure sur 24 heures	00, 01, ..., 23
%M	Minutes	00, 01, ..., 59

Utilisation du module dateutil

- Le module **dateutil** fournit des extensions du module datetime. Ce module apporte des facilités pour travailler avec les dates.
- D'abord il faut installer le module dateutil:
pip install python-dateutil
- La méthode qui nous intéresse est `dateutil.parser.parse`

Utilisation du module dateutil

- Exemple de conversion en un objet date:

```
from dateutil import parser
x=input("saisir ta date de naissance :")

y=parser.parse(x).date()
print(y)
```

La différence entre deux dates

- Si on fait la différences entre deux variables datetime, on obtient la différence en jours , heures, minutes, secondes et millisecondes, si on veut savoir le nombre d'année, il faudra faire des calculs , par exemple diviser le nombre de jours sur 365 nous permet d'obtenir la différence en année approximatif.

Exemple de calcul d'âge

- Voici un code permettant d'afficher l'âge approximatif d'une personne :

```
import datetime
x=input("Donner votre date de naissance jj-mm-aaaa")
d2=datetime.datetime.strptime(x,"%d-%m-%Y")
d1=datetime.datetime.now()
diff=d1-d2
nbjours=diff.days
age=nbjours/365
print("votre age approximatif :",round(age,0), " Années ")
```

- On peut juste faire la différence des années comme suit:

```
import datetime
x=input("Donner votre date de naissance jj-mm-aaaa")
d2=datetime.datetime.strptime(x,"%d-%m-%Y")
d1=datetime.datetime.now()
age=d1.year-d2.year
print("votre age approximatif :",age, " Années ")
```


Utilisation de la classe date

- Créer une date à l'aide de la classe date:

```
import datetime
x=datetime.date(2018,5,18)
print(x)
```

- Récupérer la date d'aujourd'hui:

```
dateDuJour=datetime.date.today()
print(dateDuJour)
```

- Saisir une date via la classe date:

```
d=input("saisir une date")
dt=datetime.date.fromisoformat(d)
print(dt)
```