

Python Datetime

Textbook

Python Datetime



Python libraries also have a certain module called [datetime](#). This allows us to work with dates and times.

To access the [datetime](#) module, use the code `import datetime`.

Now

To access the date and time for the very moment you hit "run," use the following code.

```
1 import datetime
2 x = datetime.datetime.now()
3 print(x)
```

Try it!

This will return a value that looks something like this.

```
2022-03-16 09:43:54.230000
```

You can see that this is a very specific value for the current time! It gives us the time down to a fraction of a second!

Usually, the programmer doesn't need that kind of detail. Let's explore some ways to simplify this information and make it more useful.

Date as a String

The date isn't a data type in itself, so we need to change it to a string so that we can use it in our programs. We use the following code to change the date to a string.

```
strftime( )
```

The `str` is for string and the `f` is for format.

Inside the parentheses, we add information to designate what part of the date we want to see. Continue on in the lesson to see what to put in the parentheses.

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime( ))
6
```

Today

To generate today's date as a string, displayed with numbers and dashes, use the following code.

```
datetime.date.today()
```

This will return a string that contains today's year, day, and month separated by dashes.

```
1 import datetime
2
3 x = datetime.date.today()
4
5 print(x)
```

Try it!

This lesson was written on March 16, 2022, so this would return the string `2022-03-16` .

Local Time

To generate the time down to the second as a string, use the following code.

```
"%X"
```

This will return the time down to the second as a string. Make sure the `x` is capitalized.

```
1 import datetime
```

```
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("%X"))
```

Try it!

This lesson was written at 11:19 am, so this would return the string `11:19:23` .

2 Digit Month

To generate a 2 digit month as a string, use the following code.

```
"%m"
```

This will return the month as a 2 digit string.

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("%m"))
```

Try it!

This lesson was written in the month of March, so this would return the string `03` .

2 Digit Day

To generate a 2 digit day as a string, use the following code.

```
"%d"
```

This will return the day as a 2 digit string.

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("%d"))
```

Try it!

This lesson was written on March 16, so this would return the string `16` .

4 Digit Year

To generate a 4 digit year as a string, use the following code.

```
"%Y"
```

This will return the year as a 4 digit string. Make sure the Y is capitalized.

```
1 import datetime
2
3 x = datetime.datetime.now()
```

```
4  
5 print(x.strftime("%Y"))
```

Try it!

This lesson was written in 2022, so this would return the string `2022`.

Further Date Methods

The datetime method has many ways you can pull the information. The table below shows some of the common ones.

Code	Description	Example
%a	Weekday short	Thurs
%A	Weekday long	Thursday
%b	Month short	Sept
%B	Month long	September
%y	2 Digit Year	22
%M	Minute	35
%S	Second	30

strftime() vs strptime()

In this lesson, we learned the strftime() method. There is another method referred to as strptime(). We won't get into the details of how strptime() works, but it's important to remember that the strftime() formats the date and the strptime() converts a string into a date format.

The f in strftime() stands for format.

Checkpoint

Python Datetime

Practice using the datetime library!

Import the datetime library

Create a variable named `x` and set it equal to `datetime.datetime.now()`.

Create a variable named `today` and set it equal to `datetime.date.today()`.

Using those variables, print the following:

- Now (date, time down to the decimal seconds)
- The date today (as a date with hyphens xxxx-xx-xx)
- Local Time
- 2 Digit Day
- 2 Digit Month

- 4 Digit Year

You should have **6 print statements total** for this checkpoint.

Requirements:

- import datetime library
- Create a variable named x and set it equal to datetime.datetime.now().
- Create a variable named today and set it equal to datetime.date.today().
- Print now
- Print the date today
- Print the local time
- Print the 2 digit day
- Print the 2 digit month
- Print the 4 digit year

Questions (6)

1. The code strftime() returns the date as what kind of data type?

MULTIPLE CHOICE

Choose the correct answer:

- A. string
- B. integer
- C. boolean
- D. date

2. Which of the following codes would be used to get a 2 digit month?

MULTIPLE CHOICE

Choose the correct answer:

- A. %m
- B. %d
- C. %Y
- D. %X

3. Which of the following codes would be used to get a 4 digit year?

MULTIPLE CHOICE

Choose the correct answer:

- A. %m
- B. %d
- C. %Y
- D. %X

4. Which of the following codes changes the date to a string that programmers can use?

MULTIPLE CHOICE

Choose the correct answer:

- A. strftime()
- B. strtime()
- C. strdate()
- D. str()

5. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("m"))
```

6. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 import datetime
2
3 x = datetime.now()
4
5 print(x.strftime("%d"))
```

Challenges (5)

1. Birthday Countdown!

Create a countdown until your birthday month!

Create one input that asks the user for their birth month as a 1 or 2 digit number.

The program will then print out the following with the correct values in place for x:

```
You have x months and x days until it's your birthday month!
```

For these examples, assume that current month is March.

For example:

Input (user birth month): 10

Output: You have 7 months and 10 days until it's your birthday month!

Another example:

Input: 2

Output: You have 11 months and 10 days until it's your birthday month!

IMPORTANT: This challenge is not autograded since students will be working on it during different days of the year. See if you can still get the correct output!

Hint: April, June, September, and November all have 30 days. February has 28 days. Every other month has 31 days.

Hint: Don't forget to convert your data types. The date values are strings and to do math with them, convert them to integers with `int()`.

2. New Years Resolution Check In

Many people make goals, plans, or resolutions on January 1. Write a program that prints out how many months and how many days have passed since January 1. Ask the user how their goal is going?

For example, if today was March 21, the program would print out:

```
It has been 3 months and 21 days since your New Years resolution. How are you doing?
```

Import datetime

Create a variable named x and set it equal to `datetime.datetime.now()`

Using the `%m` and the `%d` abilities, print the example sentence with the appropriate values.

Requirements:

- import datetime library
- Create a variable named x and set it equal to `datetime.datetime.now()`.
- Using `%m` and `%d`, print the example sentence with the appropriate values involved.

3. The Age of Programming

The first computer program ever created is thought to have happened in 1972. This program was simple and printed out the phrase "Hello World." It might seem like a simple program to you now, but it was a huge deal when the computer could finally do this!

Just think how far we have come since that first program!

Create a program that calculates how many years have passed since that first program.

Import `datetime`

Create a variable named `x` and set it equal to `datetime.datetime.now()`

Using the `%Y` value, print the following sentence with the appropriate value in place for the blank:

```
It has been ____ years since the first computer program ever. Look how far we have come!
```

Use the **format** concatenating method for your print statement.

Hint: Don't forget to convert your data types. The date values are strings. To do math with them, convert them to integers with `int()`.

Requirements:

- import datetime library
- Create a variable named `x` and set it equal to `datetime.datetime.now()`.
- Perform the appropriate math calculation to find the number of years that have passed.
- Print the required sentence with the appropriate value in the blank.

4. Is It a Leap Year?

Is the current year a leap year? There are many ways to tell. For this challenge, we will point out that 2020 was a leap year and leap years are every 4 years.

Create a program that will tell the user if the current year is a leap year or not.

Import `datetime`

Create a variable named `x` and set it equal to `datetime.datetime.now()`

If it is a leap year print the following sentence with the current word inserted.

Create a variable named `time` and set it equal to the current year minus 2020.

Create a variable named `calculation` and set it equal to the calculation to determine if the current year is a leap year or not. Use the variable named `time` and the modulus for your calculation.

```
This year ____, is a leap year
```

If it is not a leap year print the following sentence with the current word inserted.

```
This year ____, is not a leap year
```

Hint: Don't forget to convert your data types. The date values are strings and to do math with them, convert them to integers with `int()`.

Requirements:

- import datetime library
- Create a variable named `x` and set it equal to `datetime.datetime.now()`.
- Create a variable named `time` and set it equal to the current year minus 2020.
- Create a variable named `calculation` and set it equal to the calculation to determine if the current year is a leap year or not. Use the variable named `time` and the modulus for your calculation.
- Add an if statement that will print the appropriate response depending on the calculation for leap year

5. Holidays!

Many different cultures around the world celebrate holidays. Create a program that will print out a holiday greeting depending on what month it is. You might know of certain holidays that go with certain months. Depending on your background, you might be aware of, or celebrate, different holidays. Feel free to enter whatever holiday greeting you want for each month!

Import `datetime`

Create a variable named `x` and set it equal to `datetime.datetime.now()`

Create an if statement that checks which month is the current month. Depending on the month, the program will print out a different greeting.

Your program should have 1 if statement, 10 elif statements, and 1 else statement.

Requirements:

- import datetime library
- Create a variable named `x` and set it equal to `datetime.datetime.now()`.
- Create an if statement to check what month it is. Include at least 10 elif statements and an else statement to check for every month.

Answer Keys & Solutions

Checkpoint Solutions

Python Datetime

```
1 import datetime
2 x = datetime.datetime.now()
3 today = datetime.date.today()
4
5 print(x)
6 print(today)
7 print(x.strftime("%X"))
8 print(x.strftime("%m"))
9 print(x.strftime("%d"))
10 print(x.strftime("%Y"))
```

Questions

1. The code strftime() returns the date as what kind of data type?

MULTIPLE CHOICE

Correct Answer:

- | | |
|------------|-------------|
| A. string | ✓ Correct |
| B. integer | ✗ Incorrect |
| C. boolean | ✗ Incorrect |
| D. date | ✗ Incorrect |

Explanation:

The str stands for string and the f stands for formatting.

2. Which of the following codes would be used to get a 2 digit month?

MULTIPLE CHOICE

Correct Answer:

- | | |
|-------|-------------|
| A. %m | ✓ Correct |
| B. %d | ✗ Incorrect |

C. %Y

✗ Incorrect

D. %X

✗ Incorrect

Explanation:

The m stands for month.

3. Which of the following codes would be used to get a 4 digit year?

MULTIPLE CHOICE

Correct Answer:

A. %m

✗ Incorrect

B. %d

✗ Incorrect

C. %Y

✓ Correct

D. %X

✗ Incorrect

Explanation:

The Y stands for year.

4. Which of the following codes changes the date to a string that programmers can use?

MULTIPLE CHOICE

Correct Answer:

A. strftime()

✓ Correct

B. strtime()

✗ Incorrect

C. strdate()

✗ Incorrect

D. str()

✗ Incorrect

Explanation:

The str stands for string and the f stands for formatting.

5. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("m"))
```

Correct Solution:

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("%m"))
```

Explanation:

The code is missing a percentage sign.

6. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 import datetime
2
3 x = datetime.now()
4
5 print(x.strftime("%d"))
```

Correct Solution:

```
1 import datetime
2
3 x = datetime.datetime.now()
4
5 print(x.strftime("%d"))
```

Explanation:

This x variable needs two datetime words.

Challenges

1. Birthday Countdown!

Solution:

```
1 import datetime
```

```

2 x = datetime.datetime.now()
3
4 birthday = int(input("What month were you born in? Enter as a 1 or 2 digit
number."))
5
6 months = int(x.strftime("%m"))
7 if birthday > months:
8     monthsleft = birthday - months
9 elif birthday == months:
10    monthsleft = 0
11 else:
12    monthsleft = 12 - (months - birthday)
13
14 days = int(x.strftime("%d"))
15
16
17 if months == 4 or months == 6 or months == 9 or months == 11:
18     daysleft = (30 - days)
19 elif months == 1 or months == 3 or months == 5 or months == 7 or months == 8 or
months == 10:
20     daysleft = (31 - days)
21 else:
22     daysleft = (28 - days)
23
24
25 print(f"You have {monthsleft} months and {daysleft} days until it's your birthday
month!")

```

2. New Years Resolution Check In

Solution:

```

1 import datetime
2
3 x = datetime.datetime.now()
4
5 newyear_month = x.strftime("%m")
6 newyear_day = x.strftime("%d")
7
8 print("It has been " + newyear_month + " months and " + newyear_day + " days since
your New Years resolution. How are you doing?")

```

3. The Age of Programming

Solution:

```

1 import datetime
2 x = datetime.datetime.now()
3
4 year = int(x.strftime("%Y"))
5
6 time = year - 1972
7

```

```
8 print(f"It has been {time} years since the first computer program ever. Look how far we have come!")
```

4. Is It a Leap Year?

Solution:

```
1 import datetime
2 x = datetime.datetime.now()
3
4 year = int(x.strftime("%Y"))
5
6 time = year - 2020
7 calculation = time % 4
8
9 if calculation == 0:
10     print(f"This year {year}, is a leap year")
11 else:
12     print(f"This year {year}, is not a leap year")
```

5. Holidays!

Solution:

```
1 import datetime
2 x = datetime.datetime.now()
3
4 month = int(x.strftime("%m"))
5
6 if month == 1:
7     print("Happy New Years!")
8 elif month == 2:
9     print("Happy Valentine's Day!")
10 elif month == 3:
11     print("Happy St Patrick's Day!")
12 elif month == 4:
13     print("Happy Easter!")
14 elif month == 5:
15     print("Happy Mother's Day!")
16 elif month == 6:
17     print("Happy Father's Day!")
18 elif month == 7:
19     print("Happy 4th of July!")
20 elif month == 8:
21     print("Summertime!")
22 elif month == 9:
23     print("Back to School!")
24 elif month == 10:
25     print("Happy Halloween!")
26 elif month == 11:
27     print("Happy Thanksgiving!")
28 else:
29     print("Happy Holidays!")
```

