It is used for:
web development (server-side),
software development,
mathematics,
system scripting.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

- Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

```
C:\Users\Your Name>python helloworld.py
```

File 5

- Type the following on the Windows, Mac or Linux command line:
- C:\Users\Your Name>python
- Or, if the "python" command did not work, you can try "py":
- C:\Users\Your Name>py
- Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:
- exit()

```
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23)
[MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

- >>> print("Hello, World!")
- Hello, World!
- Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

- C:\Users\Your Name>python myfile.py

- Indentation refers to the spaces at the beginning of a code line.

- Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

- Python uses indentation to indicate a block of code.

File 9

- **Comments can be used to explain Python code.**
- **Comments can be used to make the code more readable.**
- **Comments can be used to prevent execution when testing code.**

File 11

Variables are containers for storing data values.

File 13

If you want to specify the data type of a variable, this can be done with casting.

File 15

File 16

- **String variables can be declared either by using single or double quotes:**

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)
- A variable name cannot be any of the Python keywords.

File 19

File 20

File 21

File 22

File 23

File 24

File 25

- Variables that are created outside of a function (as in all of the examples above) are known as global variables.
- Global variables can be used by everyone, both inside of functions and outside.

- If you create a variable with the same name inside a function, this variable will be local, and can only be used inside the function. The global variable with the same name will remain as it was, global and with the original value.

File 27

برای تبدیل کردن متغیر local به global :

۱)اول از کلمه کلیدی global همراه با نام متغیر استفاده کنید و

۲)سپس برای فراخوانی متغیر از نام تابع سپس نام متغیر استفاده می کنیم .

File 28

- 23e3
- 23E3

File 30

File 31

- python does not have a random() function to make a random number, but Python has a built-in module called random that can be used to make random number.
- In our Random Module Reference you will learn more about the Random module.

File 33

- **You can assign a multiline string to a variable by using three quotes:**

File 35

File 36

File 37:

توجه داشته باشید که حرفM و حرف mدر این
تمرین نتایج متفاوتی ایجاد می کنند .

File 38:

توجه کنید که اگر در negative slicing شما ابتدای باز که نقطه شروع است را به اشتباه عددی بزرگتر از انتهای بازه بگذارید مثلا

[-4:-2]

اروری دریافت نخواهید کرد اما چیزی هم نمایش داده نمی شود.

- **Whitespace is the space before and/or after the actual text**

File 40:

Learn more about String Methods with our String Methods Reference

حواست باشه یهو برنداری اینجوری بنویسی

Strip(name)

باید اینجوری بنویسی

Name.strip()

- To concatenate, or combine, two strings you can use the + operator.

- As we learned in the Python Variables chapter, we cannot combine strings and numbers with +
- But we can combine strings and numbers by using the format() method!

File 43

- **To insert characters that are illegal in a string, use an escape character.**

File 45

File 46

File 47:

حواست باشه که نقطه پایان باید ۵ باشه یعنی یه
دونه بیشتر چون کاراکتر آخر را نشان نمی دهد .

File 48

File 49

Operator	Description	Try it
()	Parentheses	Try it »
**	Exponentiation	Try it »
+x -x ~x	Unary plus, unary minus, and bitwise NOT	Try it »
* / // %	Multiplication, division, floor division, and modulus	Try it »
+ -	Addition and subtraction	Try it »
<< >>	Bitwise left and right shifts	Try it »
&	Bitwise AND	Try it »
^	Bitwise XOR	Try it »
|	Bitwise OR	Try it »
== != > >= < <= is is not in not in	Comparisons, identity, and membership operators	Try it »
not	Logical NOT	Try it »
and	AND	Try it »
or	OR	

File 50:

توجه کنید که برای is هم مقدار وهم class باید باهم برابر باشند .

File 51

File 52

- Python Collections (Arrays)
- There are four collection data types in the Python programming language:

- List is a collection which is ordered and changeable. Allows duplicate members.
- Tuple is a collection which is ordered and unchangeable. Allows duplicate members.
- Set is a collection which is unordered, unchangeable*, and unindexed. No duplicate members.
- Dictionary is a collection which is ordered** and changeable. No duplicate members.
- *Set items are unchangeable, but you can remove and/or add items whenever you like.

- **As of Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

File 54

File 55

File 56

File 57

File 58

File 59

File 60

```
newlist = [expression for item in iterable if condition == True]
```

File 62

File 63

- You cannot copy a list simply by typing list2 = list1, because: list2 will only be a reference to list1, and changes made in list1 will automatically also be made in list2.

- There are ways to make a copy, one way is to use the built-in List method copy().

File 65

File 66

File 67

File 68

File 69

- When we create a tuple, we normally assign values to it. This is called "packing" a tuple:

- But, in Python, we are also allowed to extract the values back into variables. This is called "unpacking":

File 71:

توجه کنید که ستاره asterisk پشت نا م متغیر
قرار می گیرد : x* درست نیست *x درست است .

File 72

File 73

File 74

File 75

File 76

File 77

- **Note: If the item to remove does not exist, remove() will raise an error.**

- **Note: If the item to remove does not exist, discard() will NOT raise an error.**

- The union() and update() methods joins all items from both sets.
- union() = |
- The intersection() method keeps ONLY the duplicates.
- intersection() =&
- The difference() method keeps the items from the first set that are not in the other set(s).
- difference()=-
- The symmetric_difference() method keeps all items EXCEPT the duplicates.
- symmetric_difference()=^

*all of sign just can be on sets and functions can be both on set and tuple.

File 80

- **Dictionaries are used to store data values in key:value pairs.**

File 82

- **The list of the keys is a view of the dictionary, meaning that any changes done to the dictionary will be reflected in the keys list.**

File 84

File 85

File 86

File 87

File 88

File 89

- Equals: a == b
- Not Equals: a != b
- Less than: a < b
- Less than or equal to: a <= b
- Greater than: a > b
- Greater than or equal to: a >= b
- An "if statement" is written by using the if keyword.
- The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition".
- The else keyword catches anything which isn't caught by the preceding conditions.
- The and keyword is a logical operator, and is used to combine conditional statements:
- The or keyword is a logical operator, and is used to combine conditional statements:
- The not keyword is a logical operator, and is used to reverse the result of the conditional statement:

- **One line if else statement:**

- **a = 2**
- **b = 330**
- **print("A") if a > b else print("B")**
- **This technique is known as Ternary Operators, or Conditional Expressions.**

- if statements cannot be empty, but if you for some reason have an if statement with no content, put in the pass statement to avoid getting an error.

File 93

File 94

File 95

File 96:

With the else statement we can run a block of code once when the condition no longer is true

- The while loop requires relevant variables to be ready, in this example we need to define an indexing variable, i, which we set to 1.

File 98

- **Note: The else block will NOT be executed if the loop is stopped by a break statement.**

- **Range(startpoint,endpoint,step)**

File 101

- An iterator is an object that contains a countable number of values.
- An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.
- Technically, in Python, an iterator is an object which implements the iterator protocol, which consist of the methods __iter__() and __next__().

- **Lists, tuples, dictionaries, and sets are all iterable objects. They are iterable containers which you can get an iterator from.**

File 104

To create an object/class as an iterator you have to implement the methods `__iter__()` and `__next__()` to your object.

As you have learned in the Python Classes/Objects chapter, all classes have a function called `__init__()`, which allows you to do some initializing when the object is being created.

The `__iter__()` method acts similar, you can do operations (initializing etc.), but must always return the iterator object itself.

The `__next__()` method also allows you to do operations, and must return the next item in the sequence.

File 106

File 107

- The word "polymorphism" means "many forms", and in programming it refers to methods/functions/operators with the same name that can be executed on many objects or classes.

- An example of a Python function that can be used on different objects is the len() function.

File 109

File 110

- **Note: The length of an array is always one more than the highest array index.**

- Almost everything in Python is an object, with its properties and methods.

- A Class is like an object constructor, or a "blueprint" for creating objects.

File 113

- **To understand the meaning of classes we have to understand the built-in __init__() function.**

- **All classes have a function called __init__(), which is always executed when the class is being initiated.**

- **Use the __init__() function to assign values to object properties, or other operations that are necessary to do when the object is being created**
- **Note: The __init__() function is called automatically every time the class is being used to create a new object.**

File 115

- The __str__() function controls what should be returned when the class object is represented as a string.

- If the __str__() function is not set, the string representation of the object is returned:

File 117

- Objects can also contain methods. Methods in objects are functions that belong to the object.

File 119

- Note: The self parameter is a reference to the current instance of the class, and is used to access variables that belong to the class.
- It does not have to be named self , you can call it whatever you like, but it has to be the first parameter of any function in the class

File 121

File 122

File 123

- Inheritance allows us to define a class that inherits all the methods and properties from another class.

- Parent class is the class being inherited from, also called base class.

- Child class is the class that inherits from another class, also called derived class.

File 125:

Note: Use the pass keyword when you do not want to add any other properties or methods to the class

Note: The child's __init__() function overrides the inheritance of the parent's __init__() function.

File 127

File 128

File 129

- **A variable is only available from inside the region it is created. This is called scope.**
- **A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.**

- **Function Inside Function**
- **As explained in the example above, the variable x is not available outside the function, but it is available for any function inside the function**
- **A variable created in the main body of the Python code is a global variable and belongs to the global scope.**

- **Global variables are available from within any scope, global and local.**

- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.
- Information can be passed into functions as arguments.
- Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

- the terms parameter and argument can be used for the same thing: information that are passed into a function.

- From a function's perspective:

- A parameter is the variable listed inside the parentheses in the function definition.

- An argument is the value that is sent to the function when it is called.

- By default, a function must be called with the correct number of arguments. Meaning that if your function expects 2 arguments, you have to call the function with 2 arguments, not more, and not less.

- If you do not know how many arguments that will be passed into your function, add a * before the parameter name in the function definition.

- This way the function will receive a tuple of arguments, and can access the items accordingly:

- You can also send arguments with the key = value syntax.

- This way the order of the arguments does not matter.
- The phrase Keyword Arguments are often shortened to kwargs in Python documentations.

- If you do not know how many keyword arguments that will be passed into your function, add two asterisk: ** before the parameter name in the function definition.

- This way the function will receive a dictionary of arguments, and can access the items accordingly

Keyword Arguments

```
def my_func(name, age):
    pass


my_func(name="Chetan", age=33)
```

These are **keyword arguments** as arguments are passed to function by keywords

Positional Arguments

```
def my_func(name, age):
    pass


my_func("Chetan", 33)
```

These are **positional arguments** as arguments are passed to function by position

- Python also accepts function recursion, which means a defined function can call itself.

- Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

- The developer should be very careful with recursion as it can be quite easy to slip into writing a function which never terminates, or one that uses excess amounts of memory or processor power. However, when written correctly recursion can be a very efficient and mathematically-elegant approach to programming.

File 139

File 140

File 141

File 142

File 143

File 144

- **A lambda function is a small anonymous function.**

- **A lambda function can take any number of arguments, but can only have one expression.**

- **Syntax**
- **lambda arguments : expression**

- The power of lambda is better shown when you use them as an anonymous function inside another function.

- Use lambda functions when an anonymous function is required for a short period of time.

File 147

- **Consider a module to be the same as a code library.**

- **A file containing a set of functions you want to include in your application.**

- **Note: When using a function from a module, use the syntax: module_name.function_name.**

- There is a built-in function to list all the function names (or variable names) in a module. The dir() function:
- Note: The dir() function can be used on all modules, also the ones you create yourself.

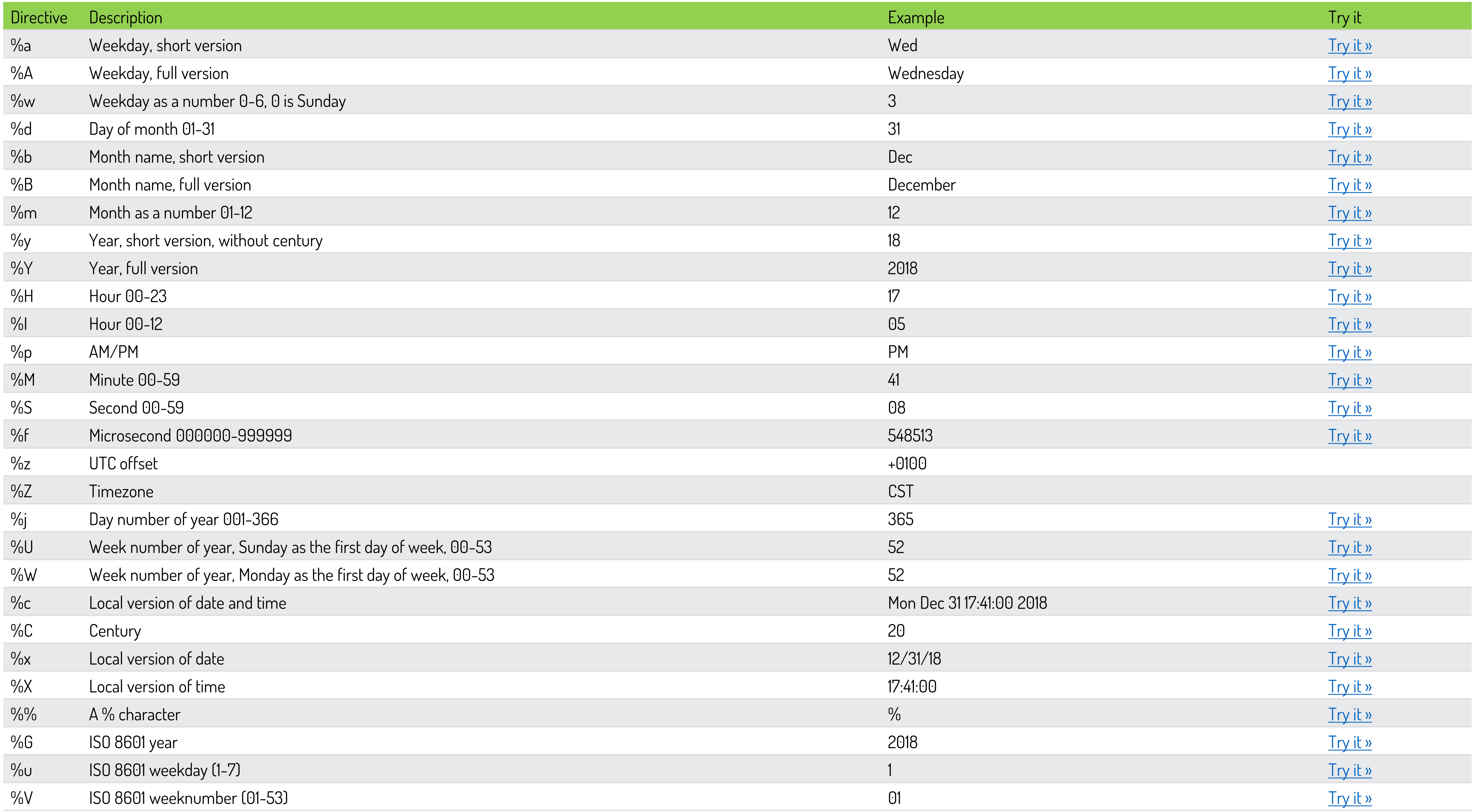
- You can choose to import only parts from a module, by using the from keyword.

- Note: When importing using the from keyword, do not use the module name when referring to elements in the module. Example: person1["age"], not mymodule.person1["age"]

File 152

- A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects.

- **The datetime object has a method for formatting date objects into readable strings.**
- **The method is called strftime(), and takes one parameter, format, to specify the format of the returned string**



Directive	Description	Example	Try it
%a	Weekday, short version	Wed	Try it »
%A	Weekday, full version	Wednesday	Try it »
%w	Weekday as a number 0-6, 0 is Sunday	3	Try it »
%d	Day of month 01-31	31	Try it »
%b	Month name, short version	Dec	Try it »
%B	Month name, full version	December	Try it »
%m	Month as a number 01-12	12	Try it »
%y	Year, short version, without century	18	Try it »
%Y	Year, full version	2018	Try it »
%H	Hour 00-23	17	Try it »
%I	Hour 00-12	05	Try it »
%p	AM/PM	PM	Try it »
%M	Minute 00-59	41	Try it »
%S	Second 00-59	08	Try it »
%f	Microsecond 000000-999999	548513	Try it »
%z	UTC offset	+0100	
%Z	Timezone	CST	
%j	Day number of year 001-366	365	Try it »
%U	Week number of year, Sunday as the first day of week, 00-53	52	Try it »
%W	Week number of year, Monday as the first day of week, 00-53	52	Try it »
%c	Local version of date and time	Mon Dec 31 17:41:00 2018	Try it »
%C	Century	20	Try it »
%x	Local version of date	12/31/18	Try it »
%X	Local version of time	17:41:00	Try it »
%%	A % character	%	Try it »
%G	ISO 8601 year	2018	Try it »
%u	ISO 8601 weekday (1-7)	1	Try it »
%V	ISO 8601 weeknumber (01-53)	01	Try it »

File 155

File 156:

In our Math Module Reference you will find a complete reference of all methods and constants that belongs to the Math module

File 157

File 158

- JSON is a syntax for storing and exchanging data.
- JSON is text, written with JavaScript object notation.
- Python has a built-in package called json, which can be used to work with JSON data.

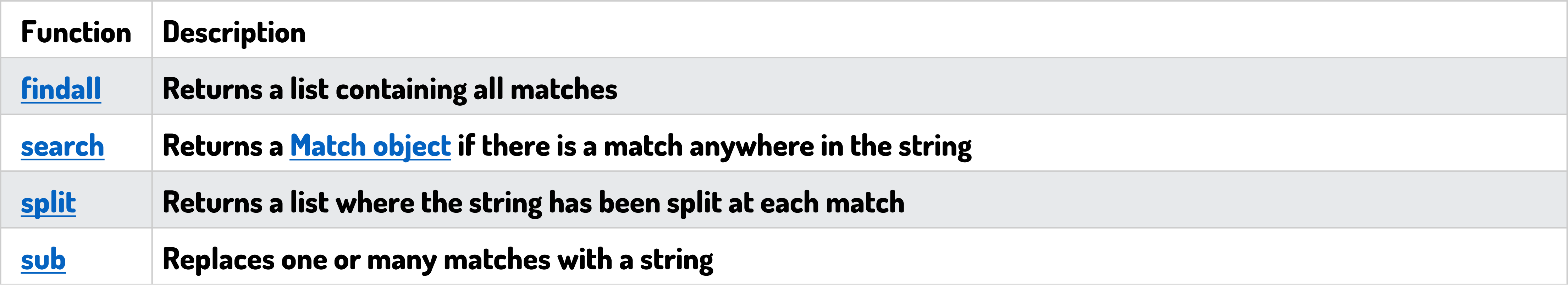
Python	JSON
dict	Object
list	Array
tuple	Array
str	String
int	Number
float	Number
True	true
False	false
None	null

File 161

- **A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.**

- **RegEx can be used to check if a string contains the specified search pattern.**

File 163



Function	Description
findall	Returns a list containing all matches
search	Returns a Match object if there is a match anywhere in the string
split	Returns a list where the string has been split at each match
sub	Replaces one or many matches with a string

Character	Description	Example	Try it
[]	A set of characters	"[a-m]"	Try it »
\	Signals a special sequence (can also be used to escape special characters)	"\d"	Try it »
.	Any character (except newline character)	"he..o"	Try it »
^	Starts with	"^hello"	Try it »
$	Ends with	"planet$"	Try it »
*	Zero or more occurrences	"he.*o"	Try it »
+	One or more occurrences	"he.+o"	Try it »
?	Zero or one occurrences	"he.?o"	Try it »
{}	Exactly the specified number of occurrences	"he.{2}o"	Try it »
|	Either or	"falls|stays"	Try it »
()	Capture and group		

Character	Description	Example	Try it
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"	Try it »
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"	Try it » Try it »
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"	Try it » Try it »
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"	Try it »
\D	Returns a match where the string DOES NOT contain digits	"\D"	Try it »
\s	Returns a match where the string contains a white space character	"\s"	Try it »
\S	Returns a match where the string DOES NOT contain a white space character	"\S"	Try it »
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"	Try it »
\W	Returns a match where the string DOES NOT contain any word characters	"\W"	Try it »
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"	Try it »

Set	Description	Try it
[arn]	Returns a match where one of the specified characters (a, r, or n) is present	Try it »
[a-n]	Returns a match for any lower case character, alphabetically between a and n	Try it »
[^arn]	Returns a match for any character EXCEPT a, r, and n	Try it »
[0123]	Returns a match where any of the specified digits (0, 1, 2, or 3) are present	Try it »
[0-9]	Returns a match for any digit between 0 and 9	Try it »
[0-5][0-9]	Returns a match for any two-digit numbers from 00 and 59	Try it »
[a-zA-Z]	Returns a match for any character alphabetically between a and z, lower case OR upper case	Try it »
[+]	In sets, +, *, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + character in the string	Try it »

File 168

File 169

File 170

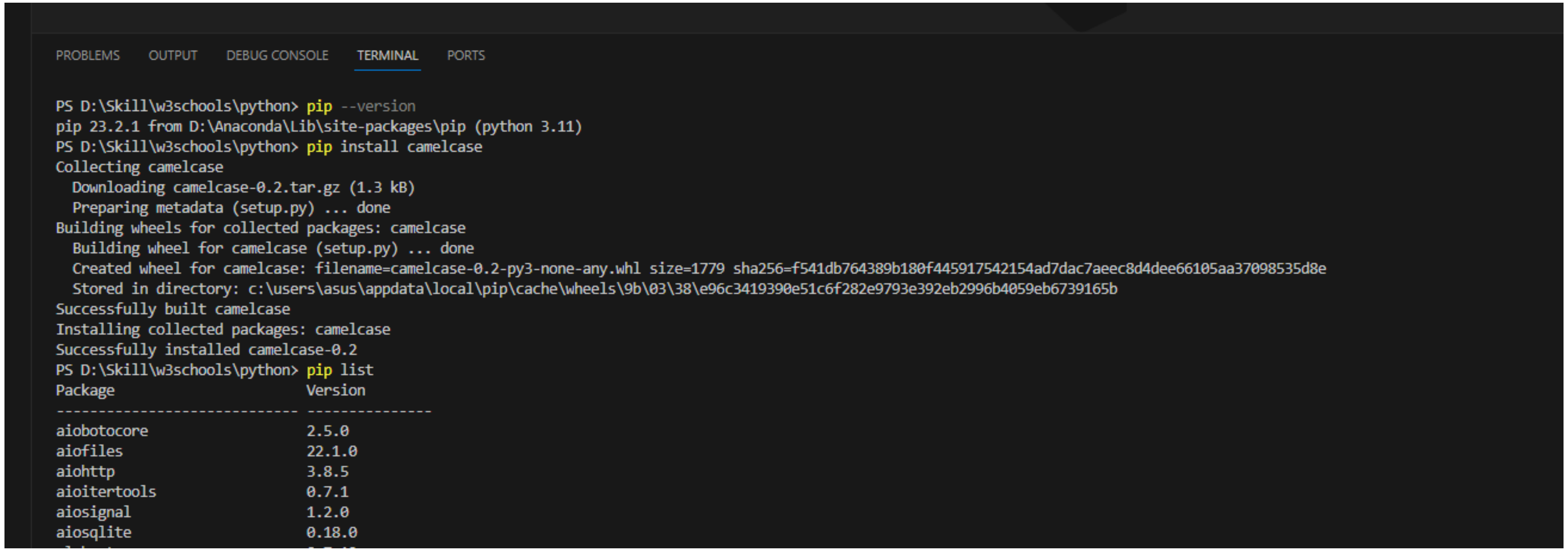
File 171

- The Match object has properties and methods used to retrieve information about the search, and the result:

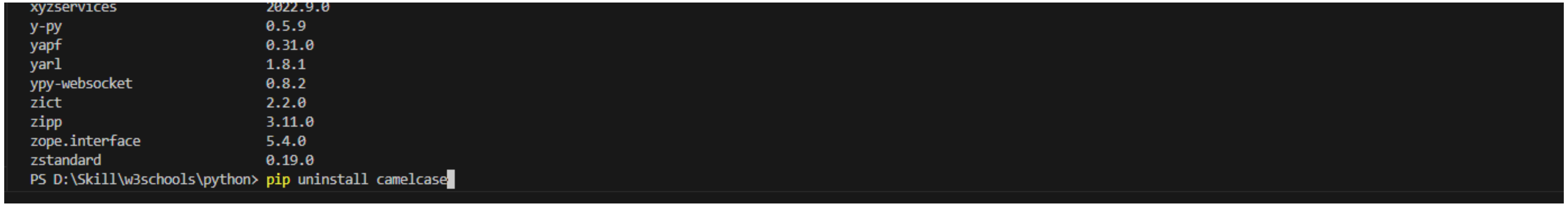
- .span() returns a tuple containing the start-, and end positions of the match.
- .string returns the string passed into the function
- .group() returns the part of the string where there was a match
- Note: If there is no match, the value None will be returned, instead of the Match Object.

- PIP is a package manager for Python packages, or modules if you like.
- A package contains all the files you need for a module.

- Modules are Python code libraries you can include in your project.



```
PS D:\Skill\w3schools\python> pip --version
pip 23.2.1 from D:\Anaconda\Lib\site-packages\pip (python 3.11)
PS D:\Skill\w3schools\python> pip install camelcase
Collecting camelcase
  Downloading camelcase-0.2.tar.gz (1.3 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: camelcase
  Building wheel for camelcase (setup.py) ... done
  Created wheel for camelcase: filename=camelcase-0.2-py3-none-any.whl size=1779 sha256=f541db764389b180f445917542154ad7dac7aeec8d4dee66105aa37098535d8e
  Stored in directory: c:\users\asus\appdata\local\pip\cache\wheels\9b\03\38\e96c3419390e51c6f282e9793e392eb2996b4059eb6739165b
Successfully built camelcase
Installing collected packages: camelcase
Successfully installed camelcase-0.2
PS D:\Skill\w3schools\python> pip list
Package                       Version
----------------------------- --------------
aiobotocore                   2.5.0
aiofiles                      22.1.0
aiohttp                       3.8.5
aioitertools                  0.7.1
aiosignal                     1.2.0
aiosqlite                     0.18.0
```



```
xyzservices                   2022.9.0
y-py                          0.5.9
yapf                          0.31.0
yarl                          1.8.1
ypy-websocket                 0.8.2
zict                          2.2.0
zipp                          3.11.0
zope.interface                5.4.0
zstandard                     0.19.0
PS D:\Skill\w3schools\python> pip uninstall camelcase
```

- The try block lets you test a block of code for errors.
- The except block lets you handle the error.
- The else block lets you execute code when there is no error.
- The finally block lets you execute code, regardless of the result of the try- and except blocks.

File 176

File 177

File 178