OCTOBER 13 – 19, 2019

المبادرة السعودية للمطورين

مسار الـPython

مشرفي المسار:

عبدالله عوده – انتصار النصار – رؤى كردي – لينا المصعبي

ملاحظات قبل بدء الدروس:

- على المتدربين نشر كل يوم الجزئية التي تم كتابتها من النص البرمجي في الـ Github تحت Topic بعنوان saudidevorg

كما تم توضيحه في دروس الـ Github سابقاً

- على المتدربين نشر كل يوم مقدار التقدم وصورة لما تم تعلمه وتطبيقه على Twitter تحت الهاشتاقات:

#المبادرة_السعودية_للمطورين

#100_يوم_برمجة

#100DaysOfCode

تمنياتنا لك بالتوفيق

المبادرة السعودية للمطورين

اليوم الخامس والخمسون

الJSON في لغة البايثون

Python JSON

JSON عبارة عن صيغة بنائية لتخزين وتبادل البيانات.

JSON is a syntax for storing and exchanging data.

➢ JSON in Python

JSONال في لغة البايثون

لغة البايثون لديها حزمة (package)مدمجة فيها تسمى JSON، والتي يمكن استخدامها للعمل مع البيانات من نوع JSON .

Python has a built-in package called json, which can be used to work with JSON data.

Example

Import the json module:

استيراد الوحدة النمطية json:

```
import json
```

➢ Parse JSON – Convert from JSON to Python

تحليل JSON - التحويل من JSON إلى بايثون

If you have a JSON string, you can parse it by using the json.loads(). Method.

The result will be a python dictionary.

إذا كان لديك نص JSON (JSON من نوع string) ، فيمكنك تحليلها باستخدام دالة. () json.loads .

ستكون النتيجة قاموس بايثون.

Example

قم بالتحويل من JSON إلى بايثون

Convert from JSON to Python:

```
import json

# some JSON:
x = '{ "name":"John", "age":30, "city":"New York"}'

# parse x:
y = json.loads(x)

# the result is a Python dictionary:
print(y["age"])
```

دالة تحليل JSON

جرب المثال

Run the example:

```
C:\Users\My Name>python demo_json.py
30
```

➢ Convert from Python to JSON

قم بالتحويل من بايثون إلى JSON

إذا كان لديك كائن Python ، يمكنك تحويله إلى نص JSON باستخدام دالة json.dumps() .

If you have a Python object, you can convert it into a JSON string by using the json.dumps() method.

Example

Convert from Python to JSON:

```
import json

# a Python object (dict):
x = {
  "name": "John",
  "age": 30,
  "city": "New York"
}

# convert into JSON:
y = json.dumps(x)

# the result is a JSON string:
print(y)
```

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_json_from_python.py
{"name": "John", "age": 30, "city": "New York"}
```

- **You can convert Pyhton objects of the following types, into JSON strings:**

يمكنك تحويل كائنات Python من الأنواع التالية ، إلى نصوص JSON:

- dict
- list
- tuple
- string
- int
- float
- true
- false
- none

- **قاموس**
- **قائمة**
- **صف**
- **نص**
- **عدد صحيح**
- **عدد نسبي**
- **القيمة المنطقية صح**
- **القيمة المنطقية خطأ**
- **لاشيء "فارغ"**

Example

Convert Python objects into JSON strings, and print the values:

تحويل كائنات بايثون إلى نصوصJSON ، ثم طباعة النتائج:

```
import json

print(json.dumps({"name": "John", "age": 30}))
print(json.dumps(["apple", "bananas"]))
print(json.dumps(("apple", "bananas")))
print(json.dumps("hello"))
print(json.dumps(42))
print(json.dumps(31.76))
print(json.dumps(True))
print(json.dumps(False))
print(json.dumps(None))
```

في هذا المثال:

أولا: قمنا باستخدام دالة json.dumps() للتحويل من بايثون إلى JSON

ثانياً: قمنا تحويل أنواع مختلفة من البيانات بالبايثون داخل دالة الطباعة **print()** لطباعة النتائج

Run the example:

```
C:\Users\My Name>python demo_json_from_python_all.py
{"name": "John", "age": 30}
["apple", "bananas"]
["apple", "bananas"]
"hello"
42
31.76
true
false
null
```

When you convert from Python to JSON, Python objects are converted into the JSON (JavaScript) equivalent:

عندما تقوم بالتحويل من Python إلى JSON ، يتم تحويل كائنات Python إلى المكافئ لها في صيغة JSON (JavaScript):

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

Example

Convert a Python object containing all the legal data types:

تحويل كائن Python يحتوي على جميع أنواع البيانات القانونية:

```
import json

x = {
  "name": "John",
  "age": 30,
  "married": True,
  "divorced": False,
  "children": ("Ann","Billy"),
  "pets": None,
  "cars": [
    {"model": "BMW 230", "mpg": 27.5},
    {"model": "Ford Edge", "mpg": 24.1}
  ]
}

# convert into JSON:
y = json.dumps(x)

# the result is a JSON string:
print(y)
```

Run the example:

```
C:\Users\My Name>python demo_json_from_python_all_in_one.py
{"name": "John", "age": 30, "married": true, "divorced": false, "children": ["Ann","Billy"], "pets": null, "cars": [{"model":
```

```
"BMW 230", "mpg": 27.5}, {"model": "Ford Edge"
```

تكملة نتيجة السطر في الشاشة ←

أتممت درسك بنجاح!

روابط قد تهمك

Useful links

JSON in Python
JSON- التعامل مع الملفات

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم السادس والخمسون

الJSON في لغة البايثون 2

Python JSON 2

➢ Format the result

تنسيق النتيجة

The example in the previous lesson prints a JSON string, but it is not very easy to read, with no indentations and line breaks.

The json.dumps() method has parameters to make it easier to read the result:

المثال في الدرس السابق يطبع نصوص JSON ، لكن ليس من السهل قراءتها ، بدون مسافات بادئة وفواصل للأسطر.

تحتوي دالة json.dumps() على معاملات لتسهيل قراءة النتيجة:

Example

Use the indent parameter to define the numbers of indents

استخدم معامل المسافة البادئة لتحديد أرقام المسافات البادئة

```
import json

x = {
  "name": "John",
  "age": 30,
  "married": True,
  "divorced": False,
  "children": ("Ann","Billy"),
  "pets": None,
  "cars": [
    {"model": "BMW 230", "mpg": 27.5},
    {"model": "Ford Edge", "mpg": 24.1}
  ]
}

# use four indents to make it easier to read the result:
print(json.dumps(x, indent=4))
```

معامل المسافة البادئة

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_json_from_python_indent.py
{
    "name": "John",
    "age": 30,
    "married": true,
    "divorced": false,
    "children": [
        "Ann",
        "Billy"
    ],
    "pets": null,
    "cars": [
        {
            "model": "BMW 230",
            "mpg": 27.5
        },
        {
            "model": "Ford Edge",
            "mpg": 24.1
        }
    ]
}
```

You can also define the separators, default value is (", ", ": "), which means using a comma and a space to separate each object, and a colon and a space to separate keys from values:

يمكنك أيضًا تحديد الفواصل، القيمة الافتراضية هي (", " ، ": ") ، مما يعني بإمكانك استخدام فاصلة ومسافة لفصل كل كائن ، ونقطتين ومسافة لفصل المفاتيح عن القيم

Example

Use the separators parameter to change the default separator:

استخدم معامل الفواصل لتغيير الفاصل الافتراضي:

```
import json

x = {
  "name": "John",
  "age": 30,
  "married": True,
  "divorced": False,
  "children": ("Ann","Billy"),
  "pets": None,
  "cars": [
    {"model": "BMW 230", "mpg": 27.5},
    {"model": "Ford Edge", "mpg": 24.1}
  ]
}

# use . and a space to separate objects, and a space, a = and a space to separate keys from their values:
print(json.dumps(x, indent=4, separators=(". ", " = ")))
```

Run the example:

جرب المثال

```
C:\Users\My Name>python demo_json_from_python_separators.py
{
    "name" = "John".
    "age" = 30.
    "married" = true.
    "divorced" = false.
    "children" = [
        "Ann".
        "Billy"
    ],
    "pets" = null.
    "cars" = [
        {
            "model" = "BMW 230".
            "mpg" = 27.5
        }.
        {
            "model" = "Ford Edge".
            "mpg" = 24.1
        }
    ]
}
```

➢ Order the result

The json.dumps() method has parameters to order the keys in the result:

تحتوي دالة json.dumps() على معاملات لترتيب المفاتيح في النتيجة:

Example

مثال

Use the sort_keys parameter to specify if the result should be sorted or not:

استخدم معامل sort_keys لتحديد ما إذا كان ينبغي فرز النتيجة أم لا:

```
import json

x = {
  "name": "John",
  "age": 30,
  "married": True,
  "divorced": False,
  "children": ("Ann","Billy"),
  "pets": None,
  "cars": [
    {"model": "BMW 230", "mpg": 27.5},
    {"model": "Ford Edge", "mpg": 24.1}
  ]
}

# sort the result alphabetically by keys:
print(json.dumps(x, indent=4, sort_keys=True))
```

في المثال التالي استخدمنا معامل الفرز لفرز النتيجة إذا كانت true

Run the example:

جرب المثال

```
C:\Users\My Name>python demo_json_from_python_sort_keys.py
{
    "age": 30,
    "cars": [
        {
            "model": "BMW 230",
            "mpg": 27.5
        },
        {
            "model": "Ford Edge",
            "mpg": 24.1
        }
    ],
    "children": [
        "Ann",
        "Billy"
    ],
    "divorced": false,
    "married": true,
    "name": "John",
    "pets": null
}
```

أتممتَ درسك بنجاح!

تابع التقدّم

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم السابع والخمسون

التعابير النمطية في لغة البايثون

Python Regular Expressions

(RegEx)

➢ What is a RegEx?

ماهي ال RegEx أو التعابير النمطية؟

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

RegEx أو التعبير العادي ، هو سلسلة من الأحرف التي تشكل نمط بحث.

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

يمكن استخدام RegEx للتحقق مما إذا كانت السلسلة تحتوي على نمط البحث المحدد.

➢ RegEx Module

Python has a built-in package called re, which can be used to work with Regular Expressions.

يحتوي Python على حزمة مضمنة تسمى re ، والتي يمكن استخدامها للعمل مع التعبيرات العادية .

```
import re
```

RegEx in Python

When you have imported the re module, you can start using regular expressions:

عند استيراد حزمة أو وحدة re ، يمكنك البدء في استخدام التعبيرات العادية:

Example

Search the string to see if it starts with "The" and ends with "Spain":

ابحث في النص لمعرفة ما إذا كان يبدأ بـ "The" وينتهي بـ:"Spain"

```
import re

#Check if the string starts with "The" and ends with "Spain":

txt = "The rain in Spain"
x = re.search("^The.*Spain$", txt)

if (x):
  print("YES! We have a match!")
else:
  print("No match")
```

في مثالنا هنا

عرفنا متغير قيمته

اسمه text وقيمته نص " the rain in spain"،

ثم استخدمنا دالة search() التابعة لحزمة re

Run the example:

جرّب المثال

```
C:\Users\My Name>python demo_regex.py
YES! We have a match!
```

➢ ReqEx Functions

The re module offers a set of functions that allows us to search a string for a match:

توفر حزمة re مجموعة من الوظائف التي تسمح لنا بالبحث في النص عن المطابقة:

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

Try one or more of these functions

جرب استخدام واحدة أو أكثر من هذه الدوال

➢ Metacharacters

أحرف التعريف

Metacharacters are characters with a special meaning

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

جرب استخدام واحدة أو أكثر من هذه الدوال

Try one or more of these functions

➢ Special Sequences المتسلسلات الخاصة

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

التسلسل الخاص هو \ متبوعًا بأحد الأحرف في القائمة أدناه ، وله معنى خاص:

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\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"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

جرب استخدام واحدة أو أكثر من هذه الدوال

Try one or more of these functions

➢ Sets

A set is a set of characters inside a pair of square brackets [] with a special meaning

المجموعة هي مجموعة من الأحرف داخل زوج من الأقواس المربعة [] ذات معنى مميز.

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

جرب استخدام واحدة أو أكثر من هذه الدوال

Try one or more of these functions

مبرمج الغد!

أتممت درسك

روابط قد تهمك

Useful links

Regular Experessions in Python for beginners

تعابير Regular expression

https://docs.python.org/3/library/re.html

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم الثامن و الخمسون

التعابير النمطية في لغة البايثون ٢

Python RegEx 2

➢ The findall() Function

دالة findall()

The findall() function returns a list containing all matches.

ترجع الدالة findall () قائمة تحتوي على جميع التطابقات.

Example

مثال

Print a list of all matches:

اطبع قائمة بجميع التطابقات.

```
import re

#Return a list containing every occurrence of "ai":

str = "The rain in Spain"
x = re.findall("ai", str)
print(x)
```

هذا المثال يوضح استخدام دالة findall() حيث أن الجزء من النص "ai" المراد البحث عنه

Run the example:

جرّب المثال

```
C:\Users\My Name>python demo_regex_findall.py
['ai', 'ai']
```

The list contains the matches in the order they are found.

If no matches are found, an empty list is returned:

تحتوي القائمة على التطابقات بالترتيب الذي تم العثور عليها به.

إذا لم يتم العثور على تطابقات ، يتم إرجاع قائمة فارغة:

Example

Return an empty list if no match was found:

إرجاع قائمة فارغة إذا لم يتم العثور على تطابق:

```
import re

str = "The rain in Spain"

#Check if "Portugal" is in the string:

x = re.findall("Portugal", str)
print(x)

if (x):
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_regex_findall2.py
[]
No match
```

➢ The search() Function دالة البحث search()

The search() function searches the string for a match, and returns a Match object if there is a match.

تقوم دالة البحث () بالبحث في النص عن المطابقة وإرجاع كائن المطابقة في حالة وجود تطابق.

If there is more than one match, only the first occurrence of the match will be returned:

إذا كان هناك أكثر من تطابق ، فسيتم إرجاع التكرار الأول فقط للتطابق:

Example

Search for the first white-space character in the string:

ابحث عن أول حرف مسافة بيضاء في النص:

```
import re

str = "The rain in Spain"
x = re.search("\s", str)

print("The first white-space character is located in position:", x.start())
```

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_regex_search.py
The first white-space character is located in position: 3
```

نتيجة تشغيل الكود

If no matches are found, the value None is returned:

إذا لم يتم العثور على تطابقات ، يتم إرجاع القيمة (لاشيء):

Example

Make a search that returns no match:

قم بإجراء بحث لا يُظهر أي تطابق:

```
import re

str = "The rain in Spain"
x = re.search("Portugal", str)
print(x)
```

Run the example:

جرّب المثال

```
C:\Users\My Name>python demo_regex_search2.py
None
```

➢ The Split() Function

دالة Split()

The split() function returns a list where the string has been split at each match:

ترجع الدالة split () قائمة حيث يتم تقسيم النص في كل تطابق:

Example

Split at each white-space character:

إقسم في كل مسافة بيضاء قدرها حرف

في المثال التالي

قمنا باستخدام دالة split() حسب المسافة البيضاء \s للنص المخزن في المتغير str

```
import re

#Split the string at every white-space character:

str = "The rain in Spain"
x = re.split("\s", str)
print(x)
```

Run the example:

جرّب المثال

```
C:\Users\My Name>python demo_regex_split.py
['The', 'rain', 'in', 'Spain']
```

أتممت درسك بنجاح!

واصل التعلّم

روابط قد تهمك

Useful links

Complete Tutorial about Python Regex

Pyhthon Regex with examples

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم التاسع والخمسون

التعابير النمطية في لغة البايثون ٣

Python RegEx 3

➢ The sub() Function

دالة sub()

The sub() function replaces the matches with the text of your choice:

تستبدل الدالة sub() التطابقات بالنص الذي تختاره:

Example

Replace every white-space character with the number 9:

يستعاض عن كل مسافة بيضاء مقدارها حرف بالرقم 9:

```
import re

#Replace all white-space characters with the digit "9":

str = "The rain in Spain"
x = re.sub("\s", "9", str)
print(x)
```

Run the example:

جرّب المثال

```
C:\Users\My Name>python demo_regex_sub.py
The9rain9in9Spain
```

You can control the number of replacements by specifying the count parameter:

يمكنك التحكم في عدد البدائل عن طريق تحديد معامل العد:

Example

Replace the first 2 occurrences of the white speace with number 9.

استبدال أول مرتين ظهور للمسافة البيضاء بالعدد ٩ :

```
import re

#Replace the first two occurrences of a white-space character with the digit 9:

str = "The rain in Spain"
x = re.sub("\s", "9", str, 2)
print(x)
```

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_regex_sub2.py
The9rain9in Spain
```

➢ Match Object كائن المطابقة

A Match Object is as object containing information about the search and the result.

كائن المطابقة هو كائن يحتوي على معلومات حول البحث والنتيجة.

ملاحظة: إذا لم يكن هناك تطابق ، فسيتم إرجاع القيمة ب(لاشيء)--none ، بدلاً من كائن المطابقة.

Example

Do a search that will return a Match Object:

قم بالبحث الذي سيعيد كائن المطابقة:

```
import re

#The search() function returns a Match object:

str = "The rain in Spain"
x = re.search("ai", str)
print(x)
```

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_regex_match.py
<_sre.SRE_Match object; span=(5, 7), match='ai'>
```

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

يحتوي كائن المطابقة على خصائص ودوال تستخدم لاسترداد معلومات حول البحث والنتيجة

.span() returns a tuple containing the start-, and end positions of the match.:

.span () تُرجع مجموعة تحتوي على مواضع البداية والنهاية للتطابق.

.string returns the string passed into the function

يقوم نوع البينات النص بإرجاع جزء من النص الذي تم التطابق فيه.

.group() returns the part of the string where there was a match

Example

Print the position (start- and end-position) of the first match occurrence.

قم بطباعة الموضع (بداية ونهاية الموقف) من حدوث التطابق الأول.

The regular expression looks for any words that starts with an upper case "S":

يبحث التعبير العادي عن أي كلمات تبدأ بالحرف العلوي:"S"

```
import re

#Search for an upper case "S" character in the beginning of a word, and print its position:

str = "The rain in Spain"
x = re.search(r"\bS\w+", str)
print(x.span())
```

Run the example:

جرّب المثال

```
C:\Users\My Name>python demo_regex_match_span.py
(12, 17)
```

Example

Print the string passed into the function:

اطبع النص الذي تم تمريره إلى الدالة.

```
import re

#The string property returns the search string:

str = "The rain in Spain"
x = re.search(r"\bS\w+", str)
print(x.string)
```

Run the example: جرّب المثال

```
C:\Users\My Name>python demo_regex_match_string.py
The rain in Spain
```

Example

Print the part of the string where there was a match.

طباعة جزء من النص حيث كان هناك تطابق.

The regular expression looks for any words that starts with an upper case "S":

يبحث التعبير العادي عن أي كلمات تبدأ بالحرف العلوي:"S"

```
import re

#Search for an upper case "S" character in the beginning of a word, and print the word:

str = "The rain in Spain"
x = re.search(r"\bS\w+", str)
print(x.group())
```

Run the example:

```
C:\Users\My Name>python demo_regex_match_group.py
Spain
```

Note: If there is no match, the value None will be returned, instead of the Match Object.

ملاحظة: إذا لم يكن هناك تطابق ، فسيتم إرجاع القيمة ب(لا شيء) أو none بدلاً من كائن المطابقة.

رائع!

أتممت درسك الأخير لهذا الأسبوع

روابط قد تهمك للاستفادة فقط

Useful links

Using Regex for Text Manipulation

Python Regular Expression Tutorial

Regular Expressions Examples

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم الستون

&

اليوم الواحد والستون

تحدي الأسبوع (يتم حله ورفعه على Github)

التحدي الأول

قم بإنشاء صف tuple و لتكن قِيَمُه هي أيام الأسبوع ثم قم بتحويله ل string باستعمال json

التحدي الثاني

باستعمال Reg Ex- regular expression- اكتب برنامج يبحث عن كلمة data في الجملة التالية data is the new oil

تحدّي إضافي - لستَ مُلزمًا بحله - :

لمحبي التحدي : اكتب برنامج يحتوي على حقائق عن لغة بايثون و استعمل دوال ال Reg Ex او JSONاو كلاهما ان أحببت

موفق دومًا

انتظرنا في دروس الأسبوع القادم