

Coding seminar

Lesson 1: Hello, World!

Ikue Hirata

Contents

What is/why Python?

Jupyter: Coding interface

Syntax – grammars

Variables

Types – int, float, string, boolean

Arithmetic

Control flows/loops

Functions

What is/why Python?

An artificial language for coding

Easy to learn/write (compared to other languages)

Widely used in science/engineering
(AI, deep learning, etc.)

Many libraries (= set of tools for common processes)

Free!



Python libraries

Mathematics

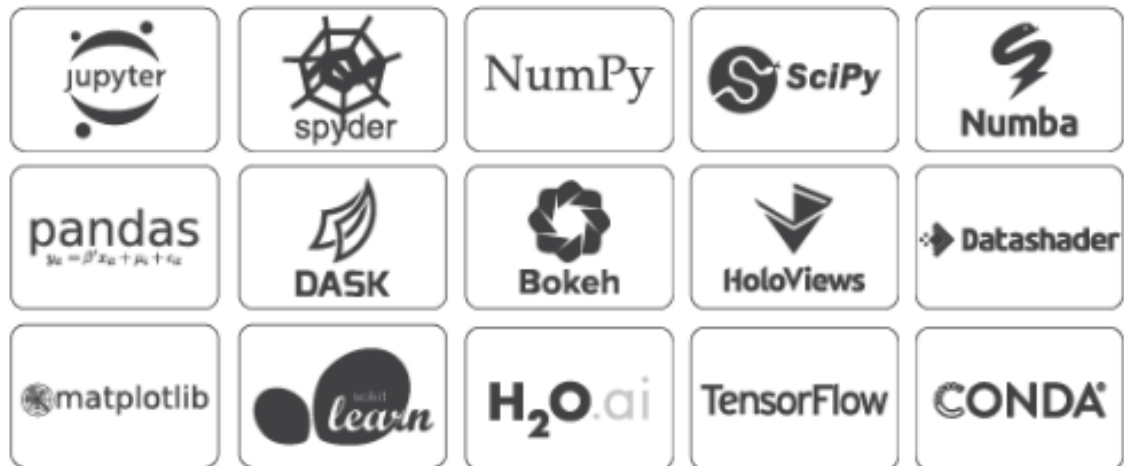
Plotting

Signal processing

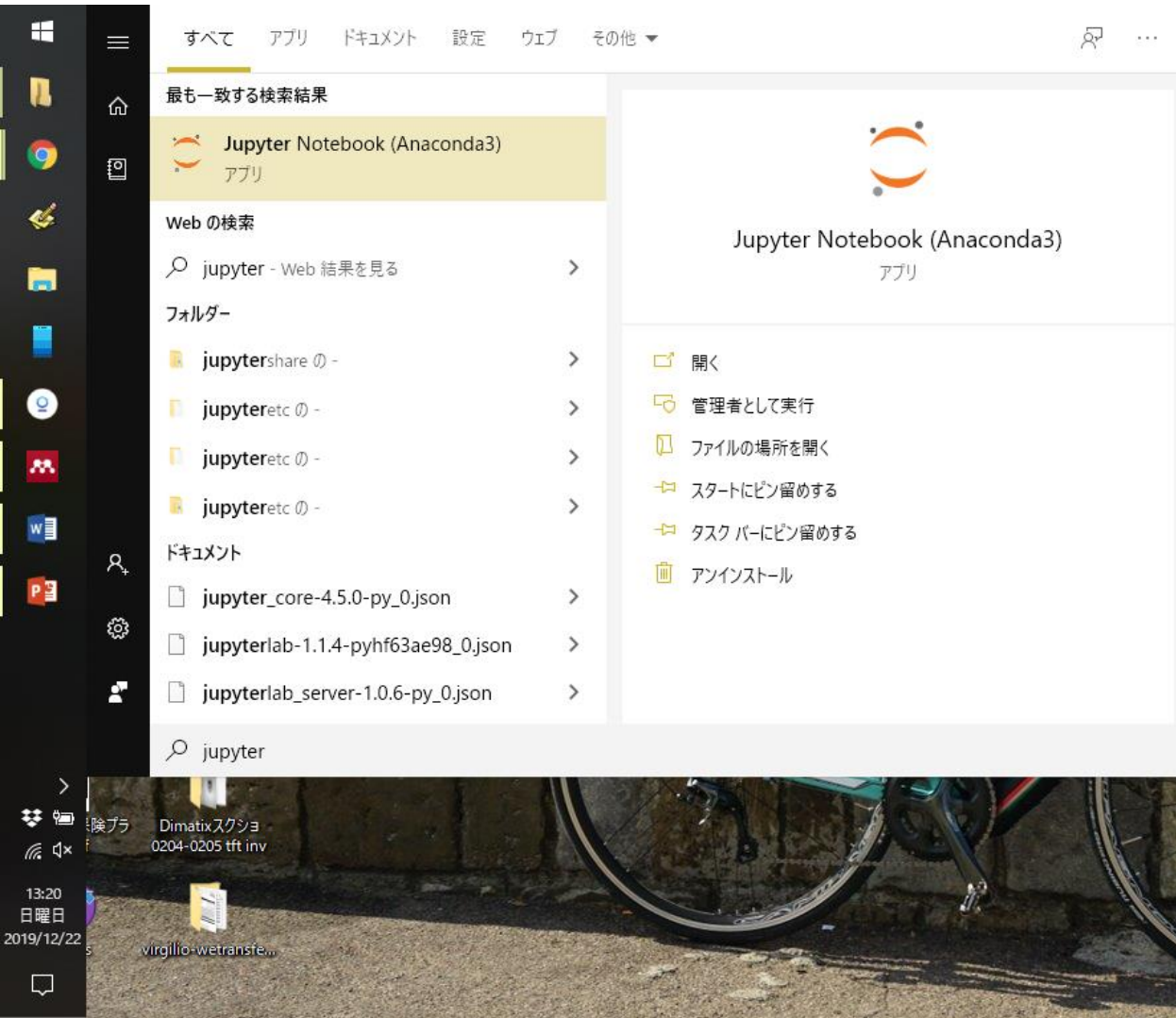
Statistics

Database

etc.



Jupyter: coding interface



Jupyter: coding interface

The screenshot shows the JupyterLab web interface in a browser window. The address bar shows 'localhost:8888/tree'. The Jupyter logo is in the top left, and 'Quit' and 'Logout' buttons are in the top right. Below the logo are tabs for 'Files', 'Running', and 'Clusters', with 'Files' being the active tab. A message says 'Select items to perform actions on them.' followed by 'Upload', 'New', and a refresh icon. A file browser table is shown with columns for selection, name, last modified, and file size. The table lists various folders like '20180530 pedot-ag printing', '3D Objects', 'Anaconda3', 'Contacts', 'Desktop', 'Documents', 'Downloads', 'Dropbox', 'eclipse', and 'eclipse workspace'.

Home × +

localhost:8888/tree

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕

<input type="checkbox"/> 0 ▾	📁 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	📁 20180530 pedot-ag printing		2年前	
<input type="checkbox"/>	📁 3D Objects		1ヶ月前	
<input type="checkbox"/>	📁 Anaconda3		1時間前	
<input type="checkbox"/>	📁 Contacts		1ヶ月前	
<input type="checkbox"/>	📁 Desktop		1時間前	
<input type="checkbox"/>	📁 Documents		1時間前	
<input type="checkbox"/>	📁 Downloads		1時間前	
<input type="checkbox"/>	📁 Dropbox		12分前	
<input type="checkbox"/>	📁 eclipse		1年前	
<input type="checkbox"/>	📁 eclipse workspace		1年前	

Create dedicated folder



Quit

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾



Name ▾

Notebook:

Python 3

Other:

Text File

Folder

Terminal

0 ▾



/



20180530 pedot-ag printing



3D Objects



Anaconda3



Contacts



Desktop

1時間前



Documents

1時間前



Downloads

1時間前



Dropbox

14分前



eclipse

1年前



eclipse workspace

1年前

Rename

[Quit](#)[Logout](#)

<input type="checkbox"/>	📁 Saved Games	1ヶ月前	
<input type="checkbox"/>	📁 Searches	1ヶ月前	
<input type="checkbox"/>	📁 thumbnails	3年前	
<input type="checkbox"/>	📁 Tracing	4年前	
<input checked="" type="checkbox"/>	📁 Untitled Folder	数秒前	
<input type="checkbox"/>	📁 userdir_v2-lkue	1年前	
<input type="checkbox"/>	📁 venvPython	3年前	
<input type="checkbox"/>	📁 Videos	1ヶ月前	
<input type="checkbox"/>	📁 winpty	1年前	
<input type="checkbox"/>	📄 apt-cyg.1	1年前	650 kB
<input type="checkbox"/>	📄 developer_key	1年前	2.38 kB
<input type="checkbox"/>	📄 seqrename.sh	2年前	366 B
<input type="checkbox"/>	📄 Sti_Trace.log	1年前	367 B

Create Python Notebook file



Quit

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾



Notebook:

Python 3

Other:

Text File

Folder

Terminal



..

The notebook list is empty.

Let's write your first code

jupyter Lesson1 (unsaved changes)



Logout

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Trusted

Python 3



Code



```
In [ ]: print("Hello, World!")
```

Ctrl + Enter: execute the code in the cell
Shift + Enter: add a cell below
Double click: edit a fixed cell

Comment

jupyter Lesson1 (autosaved)



Logout

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Trusted

Python 3



Code



```
In [1]: print("Hello, World!")
```

Hello, World!

```
In [1]: #this is a text
```

anything after # is ignored

```
In [ ]:
```

Syntax

```
print('Hello, World!')
```

Command

Options

Options can be changed

```
print('Hello, students!')
```

```
print('Hello, colleagues!')
```

```
print('Hello, Alice!')
```

```
print('Hello, Bob!')
```

```
print('Hello, Charlie!')
```

```
print('Hello, Dave!')
```

Suppose a greeting message

In [3]:

```
# greeting message  
print("Hello, students!")
```

Hello, students!

In [6]:

```
print("Hello, colleagues!")
```

Hello, colleagues!

In [7]:

```
print("Hello, Alice!")
```

Hello, Alice!

Errors

In [7]:

```
print("Hello, Eve!)
```

File "<ipython-input-7-855bf46aae91>", line 1

```
print("Hello, Eve!)
```

^

SyntaxError: EOL while scanning string literal

How to code *better*

Eliminate
redundancies

Where's redundancy?

```
print('Hello, students!')
```

```
print('Hello, colleagues!')
```

```
print('Hello, Alice!')
```

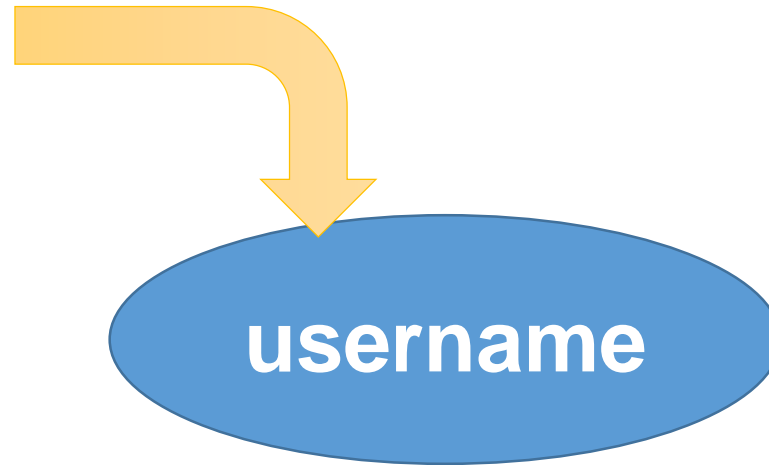
```
print('Hello, Bob!')
```

```
print('Hello, Charlie!')
```

```
print('Hello, Dave!')
```

Variables

students
colleagues
Alice
Bob
Charlie
Dave



```
print('Hello, username !')
```

Define a variable

```
username = 'Alice'
```

```
print(f'Hello, {username}!')
```

f-String

Following keywords are not permitted:

and, as, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, with, yield

Let's do it!

File Edit View Insert Cell Kernel Help

Trusted | Python 3 

```
In [16]:  
username = "Alice"  
print(f"Hello, {username}!")
```

Hello, Alice!

```
In [17]:  
username = "Bob"  
print(f"Hello, {username}!")
```

Hello, Bob!

```
In [18]:  
username = "Charlie"  
print(f"Hello, {username}!")
```

Hello, Charlie!

Variable type

int a = 1

float a = 1.2398537

bool a = True | a = False

str a = "Alice"

int/float Arithmetic

File Edit View Insert Cell Kernel Help

Trusted | Python 3 

In [24]:

```
a = 1
b = 3
c = 5
d = 11
print(a + b)
print(c - b)
print(b * c)
print(d / c)
print(d % b)
```

```
4
2
15
2.2
2
```

bool

Boolean, 'True' or 'False', without ' or '

In [26]:

```
x = True  
y = False  
print(x)  
print(y)
```

```
True  
False
```

str

File Edit View Insert Cell Kernel Help

Trusted | Python 3 

In [3]:

```
st1 = "Hello"  
st2 = ", World!"  
print(st1[0])  
print(st1[1])  
print(st1[2])  
print(st1[3])  
print(st1[4])  
print(st1 + st2)
```

H

e

l

l

o

Hello, World!


How to code *better*

Eliminate
repetitions

Make a List

Never forget **COMMAS!**

```
listofnames = ["Alice", "Bob",  
               "Charlie", "Dave"]
```



listofnames

Alice

Bob

Charlie

Dave

Use a `for/in` loop

```
In [15]: listofnames = ["Alice", "Bob", "Charlie", "Dave"]  
for username in listofnames:  
    print(f"Hello, {username}!")
```

```
Hello, Alice!  
Hello, Bob!  
Hello, Charlie!  
Hello, Dave!
```

Compound types

List

```
listofnames = ["Alice", "Bob",,,]
```

Tuple

```
group = ("Alice", "Bob",,,)
```

Dictionary

```
ages = {"Alice":14, "Bob":18,,,}
```

List

Element is associated with index
index starts from 0

```
In [19]: ic = ["Ancona", "Bolgona", "Como", "Domodossola", "Empoli"]  
ic = ic + ["Firenze", "Genova", "Hotel", "Imora"]  
print(ic[0])  
print(ic[3])  
print(ic[-1])  
print(ic[-3])
```

```
Ancona  
Domodossola  
Imora  
Genova
```

List

Part of list can be extracted by slice

`list[start:end:step]`

In [21]:

```
print(ic[1:2])  
print(ic[3:])  
print(ic[::2])  
print(ic[::-1])
```

```
['Bolgona']
```

```
['Domodossola', 'Empoli', 'Firenze', 'Genova', 'Hotel', 'Imora']
```

```
['Ancona', 'Como', 'Empoli', 'Genova', 'Imora']
```

```
['Imora', 'Hotel', 'Genova', 'Firenze', 'Empoli', 'Domodossola', 'Como', 'Bolgona', 'Ancona']
```

Dictionary

Element is associated with key

key can be non-integer

In [16]:

```
ages = {"Alice":14, "Bob":18, "Charlie":30, "Dave":90}  
print(ages["Alice"])  
print(ages["Dave"])
```

14

90

Comparison

`==` equal to

`!=` not equal to

`>` larger than

`<` smaller than

`>=` equal or larger than

`<=` equal or smaller than

Conditional statement: if

In [9]:

```
a = 1
b = 2
if a != b:
    print("Of couse 1 is not 2.")
else:
    print("Mathematical common sense has been overturned!")
```

Of couse 1 is not 2.

In [6]:

```
if "cat" == "dog":
    print("Cat is dog")
elif "cat" == "mouse":
    print("Cat is mouse")
else:
    print("Cat is not a dog nor mouse")
```

Cat is not a dog nor mouse

Control flow

for a in b : process each element in b

while A : repeat process while A is True

In [3]:

```
i = 0
while i < 5:
    print(i)
    i = i + 1
```

0
1
2
3
4

Break a control flow

In [31]:

```
h = 8
while h < 24:
    print(f"{h}.00 working")
    h = h + 3
    if h > 22:
        break
print("I'm too tired!")
```

```
8.00 working
11.00 working
14.00 working
17.00 working
20.00 working
I'm too tired!
```

User-made function

File Edit View Insert Cell Kernel Help

Trusted | Python 3 

In [25]:

```
def func():  
    print('this is a test function')  
func()
```

this is a test function

In [27]:

```
def fahrenheit(T_in_celsius):  
    # returns the temperature in degrees Fahrenheit  
    return (T_in_celsius * 9 / 5) + 32  
print(fahrenheit(25))
```

77.0

Today's Summary

Variables	int	0, 1, 2, 3...
	float	1.4396372
	bool	True/False
	str	"Alice"
Compounds	list	[0, 1, 2, 3, 4...]
	tuple	(0, 1, 2, 3, 4...)
Comparison	if A:	==, !=, >, <, >=, <=
Control flow	for A in B:	while A: break
Function	def nameOfFunction():	

Exercise 1

Branch: master ▾

CodingSeminarCMBR / Lesson1_Exercise1.ipynb

Find file

Copy path



ikuehirata exercise updated

9cdbbdb 17 seconds ago

1 contributor

50 lines (50 sloc) | 1.86 KB



Raw

Blame

History



Exercise 1

1. Print an invitation for a drink to the people in the lists following the conditions below;
2. Each person answers yes or no. Print the name of the attendee and count how many people attend.

Conditions

1. Names are stored in the list `names`, ages in `ages`, answers in `ans`;
2. Do not invite underaged (less than 16.)

```
In [1]: names = ["Alice", "Bob", "Charlie", "Dave", "Eve", "Faythe", "Grace", "Heidi", "Ivan", "Judy", "Mallory", "Niaj", "Olivia", "Peggy", "Rupert", "Sybil", "Trudy", "Victor", "Walter"]
ages = {'Alice': 58, 'Bob': 10, 'Charlie': 27, 'Dave': 12, 'Eve': 50, 'Faythe': 35, 'Grace': 39, 'Heidi': 23, 'Ivan': 53, 'Judy': 28, 'Mallory': 49, 'Niaj': 33, 'Olivia': 17, 'Peggy': 38, 'Rupert': 55, 'Sybil': 9, 'Trudy': 40, 'Victor': 56, 'Walter': 37}
ans = {'Alice': 'Yes', 'Bob': 'No', 'Charlie': 'No', 'Dave': 'Yes', 'Eve': 'No', 'Faythe': 'No', 'Grace': 'No', 'Heidi': 'Yes', 'Ivan': 'Yes', 'Judy': 'Yes', 'Mallory': 'No', 'Niaj': 'No', 'Olivia': 'No', 'Peggy': 'Yes', 'Rupert': 'No', 'Sybil': 'Yes', 'Trudy': 'Yes', 'Victor': 'No', 'Walter': 'No'}
# invitation message example: "Hello Alice, let's grab some beer!"
```

Exercise 2

ikuehirata / CodingSeminarCMBR

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Branch: master

CodingSeminarCMBR / Lesson1_Exercise2.ipynb

Find file

Copy path

ikuehirata add l1-e2

76a71b1 1 minute ago

1 contributor

60 lines (60 sloc) | 1.27 KB

<>

File icon

Raw

Blame

History

Monitor icon

Edit icon

Delete icon

Lesson 1 - Exercise 2

Write a code that determines if the input number is a primal number or not, such as:

```
def isprime(n):  
    your code here  
    if (n is a prime number):  
        return True  
    else:  
        return False  
  
isprime(10)  
isprime(297)  
isprime(104593)
```

The output is False, ?, ? (find the answers by yourself!)

Your friend: official documentation

The screenshot shows a web browser window with the address bar displaying `docs.python.org/3.7/`. The page title is "3.7.6 Documentation". The main content area is titled "Python 3.7.6 documentation" and includes a welcome message: "Welcome! This is the documentation for Python 3.7.6." Below this, there is a section "Parts of the documentation:" with links to "What's new in Python 3.7?", "Tutorial", "Library Reference", "Language Reference", "Python Setup and Usage", and "Python HOWTOs". To the right of these links are links to "Installing Python Modules", "Distributing Python Modules", "Extending and Embedding", "Python/C API", and "FAQs". At the bottom, there is a section "Indices and tables:" with a link to "Global Module Index". On the left side of the page, there is a sidebar with "Download" (a link to "Download these documents"), "Docs by version" (a list of Python versions from 3.9 to 2.7), and "Other resources" (links to "PEP Index", "Beginner's Guide", "Book List", and "Audio/Visual Talks"). The top of the page has a navigation bar with "Python", "English", "3.7.6", and "Documentation" dropdowns, a "Quick search" box, and links to "modules" and "index".

3.7.6 Documentation

docs.python.org/3.7/

Python » English 3.7.6 Documentation » Quick search Go | modules | index

Download

Download these documents

Docs by version

- Python 3.9 (in development)
- Python 3.8 (stable)
- Python 3.7 (stable)
- Python 3.6 (security-fixes)
- Python 3.5 (security-fixes)
- Python 2.7 (stable)
- All versions

Other resources

- PEP Index
- Beginner's Guide
- Book List
- Audio/Visual Talks

Python 3.7.6 documentation

Welcome! This is the documentation for Python 3.7.6.

Parts of the documentation:

- [What's new in Python 3.7?](#)
or all "What's new" documents since 2.0
- [Tutorial](#)
start here
- [Library Reference](#)
keep this under your pillow
- [Language Reference](#)
describes syntax and language elements
- [Python Setup and Usage](#)
how to use Python on different platforms
- [Python HOWTOs](#)
in-depth documents on specific topics

Indices and tables:

- [Global Module Index](#)
quick access to all modules

- [Installing Python Modules](#)
installing from the Python Package Index & other sources
- [Distributing Python Modules](#)
publishing modules for installation by others
- [Extending and Embedding](#)
tutorial for C/C++ programmers
- [Python/C API](#)
reference for C/C++ programmers
- [FAQs](#)
frequently asked questions (with answers!)

Search page

search this documentation

Your friend: Stack Overflow

The screenshot shows a web browser window with the Stack Overflow website. The address bar shows the URL: `stackoverflow.com/questions/3996904/generate-random-integers-between-0-and-9`. The page title is "Generate random integers between 0 and 9". The question was asked 9 years, 2 months ago, is active (20 days ago), and has been viewed 1.8m times. The question text is: "How can I generate random integers between 0 and 9 (inclusive) in Python? For example, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9". The question has 1240 votes and 19 answers. The top answer, by user6269864, has 1931 votes and shows the Python code: `from random import randrange` and `print(randrange(10))`. The right sidebar contains a "Blog" section with links to "What senior developers can learn from beginners" and "WebSockets for fun and profit", a "Featured on Meta" section with links to "We're lowering the close/reopen vote threshold from 5 to 3 for good" and "Why was I just awarded a bunch of 'Announcer' badges?", and a banner for "Fachhochschule Graubünden Academic Research" with a link to "Dozent/in Daten-Visualisierung (80 - 100 %)" in Chur, Schweiz.

python - Generate random integ x +

stackoverflow.com/questions/3996904/generate-random-integers-between-0-and-9

stackoverflow Products Customers Use cases Search... Log in Sign up

Home PUBLIC Stack Overflow Tags Users Jobs TEAMS What's this? First 25 Users Free

Generate random integers between 0 and 9

Asked 9 years, 2 months ago Active 20 days ago Viewed 1.8m times

Ask Question

How can I generate random integers between 0 and 9 (inclusive) in Python?

For example, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

python random integer

1240

share improve this question 155

edited Oct 18 '18 at 8:08 user6269864

asked Oct 22 '10 at 12:48 aneurym 48.6k 88 246 456

add a comment

19 Answers

active oldest votes

Try:

1931

```
from random import randrange
print(randrange(10))
```

More info: <http://docs.python.org/library/random.html#random.randrange>

share improve this answer

edited Jul 1 at 12:11 Georgy 3,277 4 24 38

answered Oct 22 '10 at 12:51 kovshenin 25k 4 31 42

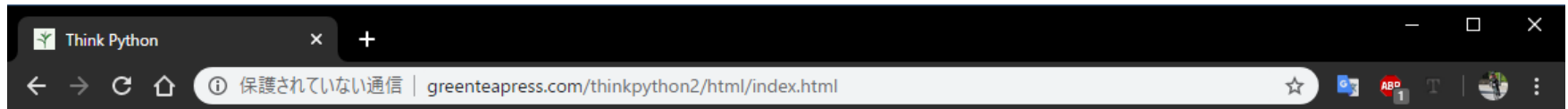
FH GR

Fachhochschule Graubünden
Academic Research

Dozent/in Daten-Visualisierung (80 - 100 %)
Chur, Schweiz

python java

Your friend: other materials



[Buy this book at Amazon.com](#)

Think Python: How to Think Like a Computer Scientist

Allen B. Downey

2nd Edition, Version 2.4.0

- [Preface](#)
 - [The strange history of this book](#)
 - [Acknowledgments](#)
 - [Contributor List](#)
- [The way of the program](#)
 - [What is a program?](#)
 - [Running Python](#)
 - [The first program](#)
 - [Arithmetic operators](#)
 - [Values and types](#)
 - [Formal and natural languages](#)
 - [Debugging](#)
 - [Glossary](#)
 - [Exercises](#)
- [Variables, expressions and statements](#)
 - [Assignment statements](#)
 - [Variable names](#)
 - [Expressions and statements](#)
 - [Script mode](#)
 - [Order of operations](#)

Contribute

If you would like to make a contribution to support my books, you can use the button below and pay with PayPal. Thank you!

Pay what you want:

Small \$1.00 USD

[Pay Now](#)

Are you using one of our books in a class?

We'd like to know about it. Please consider filling out [this short survey](#).

Think DSP



Your best friend: Google



python list index



ログイン

[すべて](#) [画像](#) [地図](#) [動画](#) [ニュース](#) [もっと見る](#) [設定](#) [ツール](#)

約 142,000,000 件 (0.29 秒)

[note.nkmk.me](#) > Python ▾

Pythonのリストの要素のインデックス（何番目か）を取得 - nkmk note

2018/07/02 - **Python**のリスト（配列）の要素のインデックス、つまりその要素が何番目に格納されているかを取得する方法を説明する。以下の内容について説明する。リストの要素が重複していない場合: `index()` リストの要素が重複している場合: `index(), ...`

[www.programiz.com](#) > methods > list ▾ [このページを訳す](#)

Python List index() - Programiz

The `index()` method searches an element in the `list` and returns its `index`.

[www.javadrive.jp](#) > Python入門 > リスト ▾

リストの要素を取得する | Python入門 - Let'sプログラミング

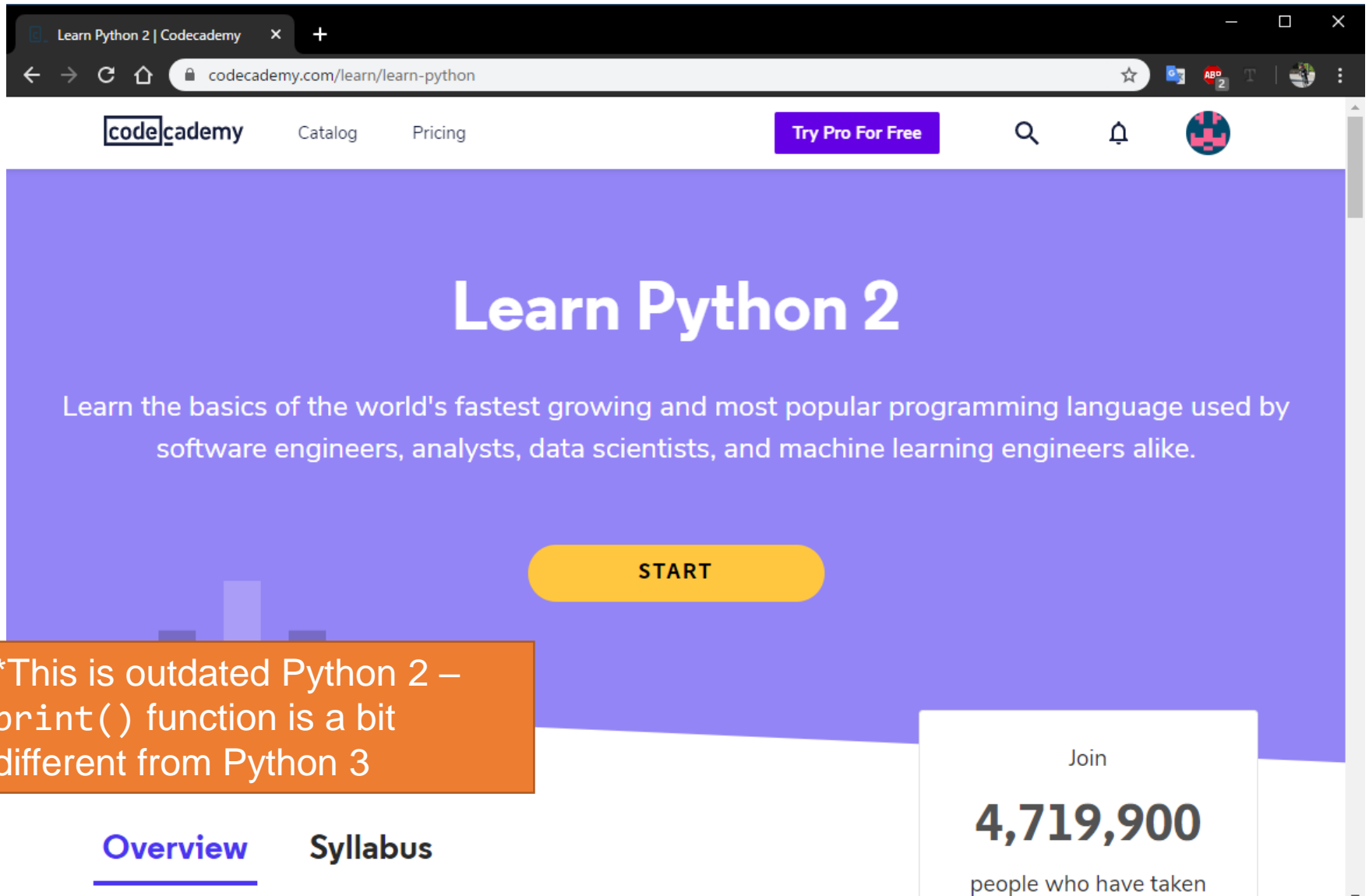
作成したリストに対してインデックスを指定することで要素を取得することができます。ここではリストの要素を取得する方法について解説します。

[www.geeksforgeeks.org](#) > python-list-index ▾ [このページを訳す](#)

Python list | index() - GeeksforGeeks

`index()` is an inbuilt function in **Python**, which searches for given element from start of the `list` and returns the lowest `index` ... element - The element whose lowest `index` will be returned. start (Optional) - The position from where the search ...

To study by yourself: codecademy



Learn Python 2 | Codecademy

codecademy Catalog Pricing Try Pro For Free

Learn Python 2

Learn the basics of the world's fastest growing and most popular programming language used by software engineers, analysts, data scientists, and machine learning engineers alike.

START

*This is outdated Python 2 – print() function is a bit different from Python 3

Overview Syllabus

Join
4,719,900
people who have taken