

BIOINFORMATICS

INTRODUCTION TO PYTHON

2023-09-04

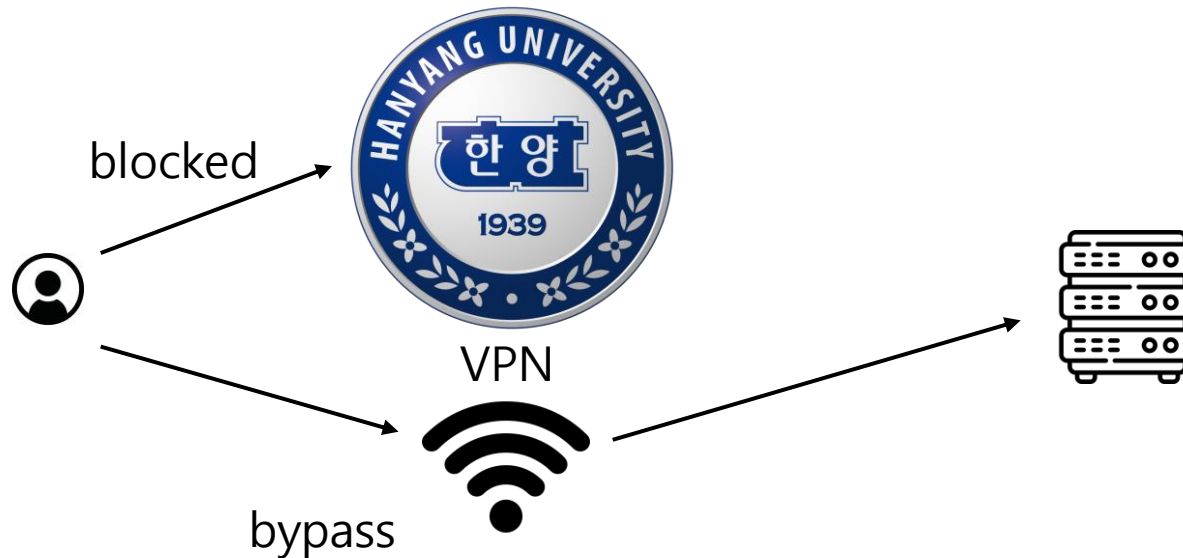
Dept. of Life Science, HYU

Class assistances

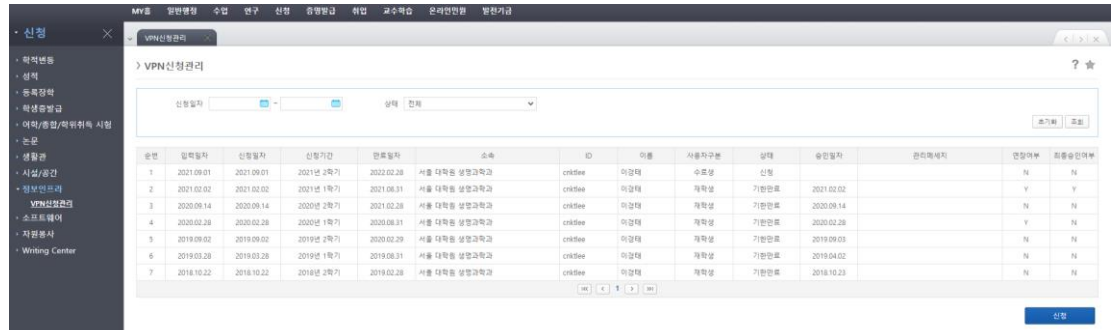
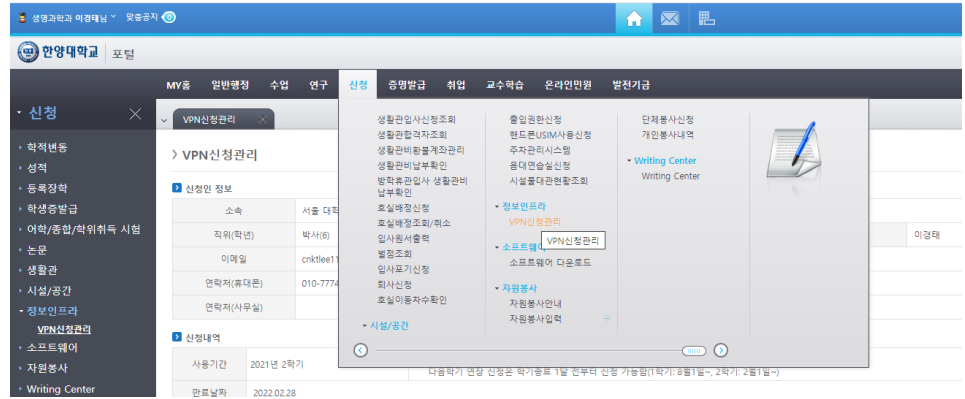
- 김현우(Hyun Woo Kim) barbaric@hanyang.ac.kr
 - 김솔빈(Solbeen Kim) solbin0737@hanyang.ac.kr
 - 이경태 박사님 (Dr. Kyungtae Lee)
-
- Weekly assignment due date&time : Sunday 23:59
 - assignment late submission (with penalty) : ~6h after due time
any further submission will be graded as 0 points
-
- Class contents will be uploaded on LMS

Apply for VPN

- 가상사설망 (Virtual Private Network)을 의미하는 **VPN은 사용자와 인터넷 간에 보안 연결을 구축합니다.** 개인 정보 보호와 익명성 유지 기능을 제공하기 때문에 다음을 수행 할 수 있습니다
- VPN을 이용하여 학교 외부에서 교내 서버에 접속 할 수 있음



신청방법



위와 같이 개인정보를 수집-이용하는데 동의하십니까 ?

필수 정보	동의함	<input checked="" type="radio"/>	동의하지 않음	<input type="radio"/>
-------	-----	----------------------------------	---------	-----------------------

관련 문의
- 신청 관련 문의 : 정보통신처 정보인프라팀(내선 1427)

1. 원격 접근 보안 서약서

본인은 상기 업무 수행을 위하여, 학내 행정 업무용 PC에 대한 원격 접속이 필요합니다. 이에, 원격접속 시 다음 사항을 준수할 것을 약속하 서약합니다.

- 나는 부여받은 인증 관련 정보를 타인에게 유출하지 아니한다.
- 나는 원격근무 중 작성, 저장, 열람, 출력한 문서는 업무 목적에만 활용하고 타인에게 유출하지 아니한다.
- 나는 원격 접속 PC 및 원격접근 대상 PC의 보안 상태를 최신으로 유지하며, 정기적인 점검 및 백신 프로그램 검사를 통하여 최신 보안 상태를 유지한다.
- 나는 개인정보보호 및 정보보호를 위한 준수사항을 성실히 이행한다.

[사용 권한신청 및 사용 안내]
교육부 및 행정부의 권고에 따라서 저희 한양대학교는 외부에서 접근하는 모든 서비스 포트가 차단되어 있습니다. 외부에서 학내로 접속이 필요한 사용자는 아래와 같이 사용안내에 따라서 VPN 사용 권한을 신청하여 주시기 바랍니다.

- 신청방법 : 원할대 현재 소속 구성원에게만 제공됩니다.(포털 계정이 있는 사용자만 신청가능)
- 사용목적 : 업무 및 연구용 목적 외에는 사용을 불허 합니다.
- 사용기간 : 최대 6개월(학기말 까지) 사용 신청 가능합니다.(사용 기간이 만료 후, 재승인 전까지 사용이 불가능합니다.)
- 사용 신청한 기간 동안만 제공되며, 자동 연장되지 않으므로 기간 만료 시 재신청 하셔야 합니다.
- 제한사항 :
 - VPN을 사용하는 목적은 개인적 목적이거나, 학교 전체의 보안정책에 위해 될 경우 승인 되지 않을 수 있습니다.
 - 사용자 단말의 관리 소홀로 인한 학내 보안에 문제가 발생할 경우 즉시 권한이 중지됩니다.

위 사항을 확인합니다.

VPN신청관리

1. 저장 후 신청까지 완료해야 정상적으로 접속이 진행됩니다.
2. 안내 바로가기: http://vpninfo.hanyang.ac.kr/Manual/VPN-Manual*onepage.pdf

2. 신청인 정보

소속:

직위(학번): 성명:

이메일:

연락처(휴대폰):

연락처(사무실):

3. 신청내역

사용기간: 2023년 2학기 최대 연장은 신청한 학기의 학기말을 넘지 못함 (1학기:~9월31일, 2학기:~2월28일)
다중학기 연장 신청은 학기종료 1달 전부터 신청 가능함(1학기: 8월1일~; 2학기: 2월1일~)

만료날짜: 2024.02.29

업무분류: 행정업무 연구업무 업무수강 기타

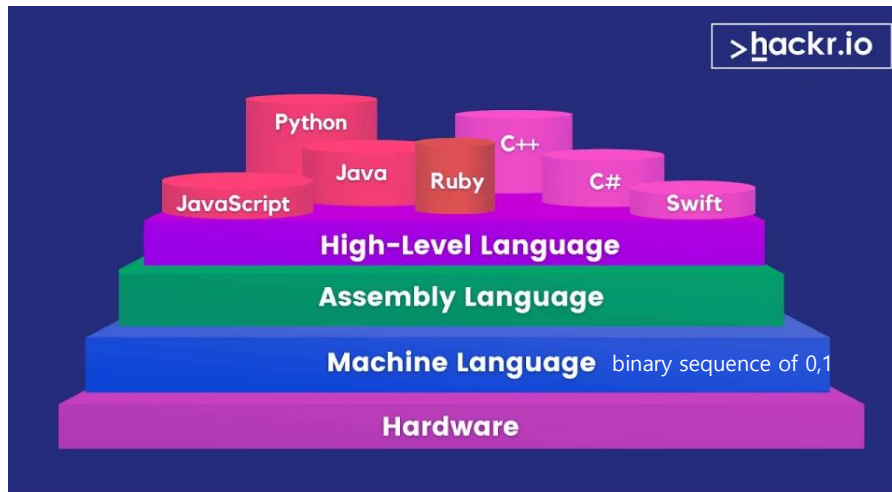
업무세부내역:

기타 요청사항:

접속 승인이 난 후에 <http://vpn.hanyang.ac.kr> 에 접속해서 사용방법 숙지 및 프로그램 설치

What is Python?

- Python is a high level interpreted language, which is best suited for writing scripts for automation and code re-usability



BASIS FOR COMPARISON	HIGH-LEVEL LANGUAGE	LOW-LEVEL LANGUAGE
Basic	Programmer amiable	Machine-friendly
Speed of execution	Fast	Slow
Translation	Requires compiler or an interpreter.	Assembler is required while machine language is directly executed.
Memory efficiency	Low	High
Comprehensibility	Understandable	Hard to understand
Portability and machine dependency	Portable and runnable in any platforms.	Non-portable and machine dependent.
Debugging and maintenance	Simple	Quite complex

hackr.io/blog/best-programming-languages-to-learn

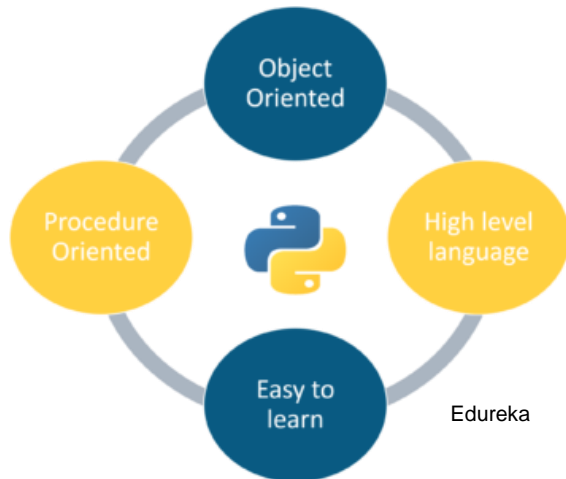
techdifferences.com/difference-between-high-level-language-and-low-level-language.html

Creation of Python

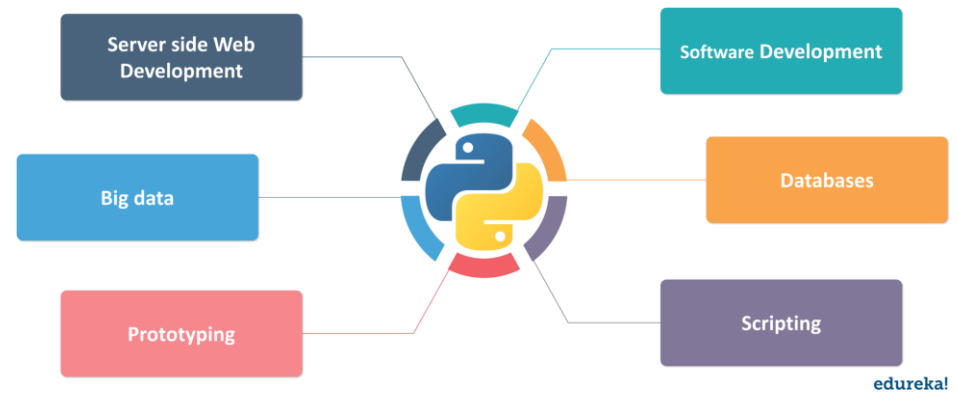
- It was created in 1991 by Guido Van Rossum



<Features of Python>



<Application of Python>



Key elements in Python – 1.Keywords

- Keywords are special names that are already present in Python. You can use these keywords for specific functionality while writing a Python program

Keywords in Python				
False	class	<u>finally</u>	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	<u>elif</u>	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Key elements in Python – 2.Variables

- Variables are like a memory location where you can store a value.

```
1 | x = 10  
2 | y = 20  
3 | name = 'edureka'
```

- In the case above, x,y, name are newly created variables.
- To declare a variable in python, you only have to assign a value to it.

Key elements in Python – 3.Data types

- Data types in Python : Numbers, String, List, Dictionary, Set, Tuple

Numbers : Numbers or numerical data type is used for numerical values.

↳ 4 types of numerical data : integer, float, complex, boolean

```
1 | #integers are used to declare whole numbers.
2 | x = 10
3 | y = 20
4 | #float data types is used to declare decimal point values
5 | x = 10.25
6 | y = 20.342
7 | #complex numbers denote the imaginary values
8 | x = 10 + 15j
9 | #boolean is used to get categorical output
10 | num = x < 5
11 | #the output will be either true or false here.
```

Key elements in Python – 3.Data types

String : is used to represent characters or alphabets. You can declare a string using single “ or double quotes “”.

```
1 | name = 'edureka'  
2 | course = "python"
```

List : is a collection where you can store different values. It need not be uniform and can have different values

```
1 | a = [10,20,30,40,50]  
2 | #append will add the value at the end of the list  
3 | a.append('edureka')  
4 | #insert will add the value at the specified index  
5 | a.insert(2,'edureka')  
6 | #reverse will reverse the list  
7 | a.reverse()  
8 | print(a)  
9 | #the output will be  
10 | ['edureka', 50, 40, 30, 'edureka', 20, 10]
```

Key elements in Python – 3.Data types

Dictionary : is unordered and changeable data structure. We use the key : value pairs in a dictionary. Since the keys are unique, we can use them as indexes to access the values from dictionary

```
1 | my_dictionary = { 'key1' : 'edureka' , 2 : 'python'}
2 | mydictionary['key1']
3 | #this will get the value 'edureka'. the same purpose can be fulfilled by get().
4 | my_dictionary.get(2)
5 | #this will get the value 'python'.
```

For example, let's make a dictionary containing the students and their student ID for 2023 bioinformatics.

```
bio_2023 = {'hyunwoo' : 2023012345, 'solbeen' : 2023012346}
```

Tuple : It is another collection which is ordered and unchangeable. We declare the tuples in Python with round brackets.

```
1 | mytuple = (10,20,30,40,50,50,50,60)
2 | mytuple.count(40)
3 | #this will get the count of duplicate values.
4 | mytuple.index(20)
5 | #this will get the index for the vale 20.
```

Key elements in Python – 4.Operators

Operators in Python are used to do operations between two values or variables

Arithmetic Operators



Assignment Operators

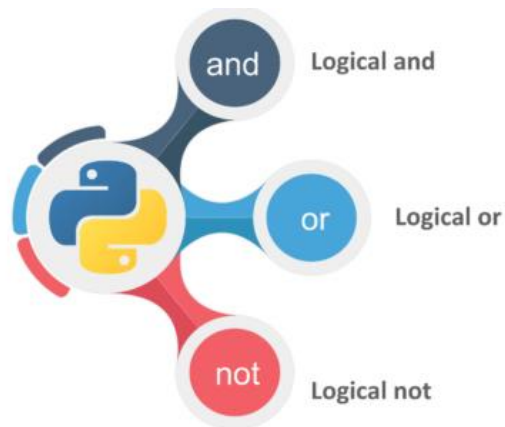
OPERATOR	EXAMPLE
=	x = 10
+=	x += 10
-=	x -= 10
*=	x *= 5
%=	x %= 20
**=	x **= 10
//=	x //= 25
=	y = 20
^=	x ^= 15
&=	x &= 12

Key elements in Python – 4.Operators

Comparison Operators



Logical Operators



Key elements in Python – 4.Operators

Membership Operators

IN

- Returns **true** if a sequence with the specified value is present in the object
- Example:
`x in y`

NOT IN

- Returns **true** if a sequence with the specified value is not present in the object
- Example:
`x not in y`

Identity Operators

IS

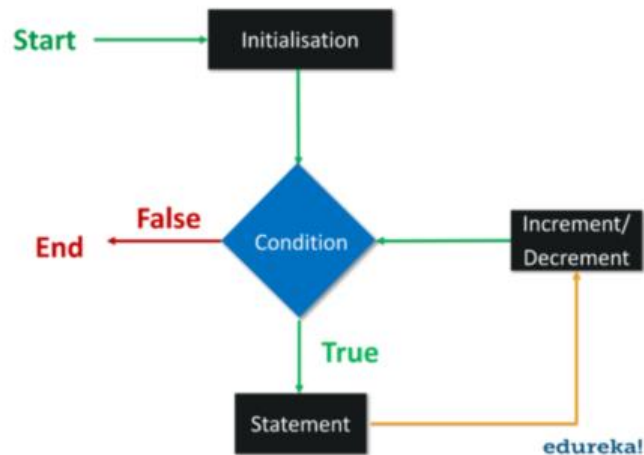
- Returns **true** if both variables are same object
- Example:
`x is y`

IS NOT

- Returns **true** if both variables are not same object
- Example:
`x is not y`

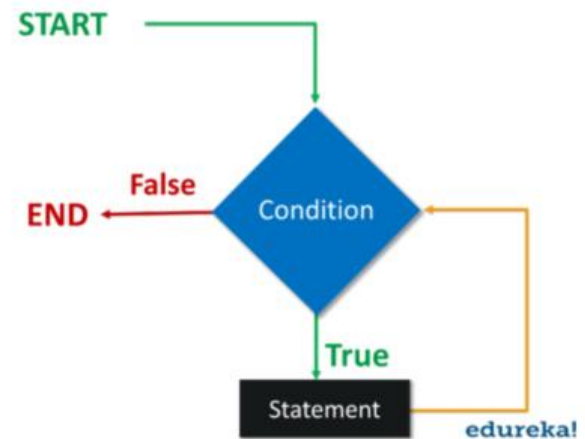
Key elements in Python – 5.Loops

<For loops>



```
1 | for x in range(10):  
2 |     print(x)
```

<While loops>



```
1 | i = 1  
2 | while i < 6:  
3 |     print(i)  
4 |     i += 1  
5 | #the output will be numbers from 1-5.
```

A “For loop” is used to execute statements once every iteration. We already know the number of iterations that are going to execute

A “While loop” executes statements as long as the condition is true. We specify the condition in the beginning of the loop and as soon as the condition is false, the execute stops.

Key elements in Python – 6.Conditional and Control statements

Conditional statements :

```
x =10
if x > 5:
    print('greater')
elif x == 5:
    print('equal')
else:
    print('smaller')
```

Control statements : are used to control the flow of execution in the program

break

```
1 name = 'edureka'
2 for val in name:
3     if val == 'r':
4         break
5     print(i)
6 #the output will be
7 e
8 d
9 u
```

Continue

```
1 name = 'edureka'
2 for val in name:
3     if val == 'r':
4         continue
5     print(i)
6 #the output will be
7 e
8 d
9 u
10 e
11 k
12 a
```

Pass

```
1 name = 'edureka'
2 for val in name:
3     if val == 'r':
4         pass
5     print(i)
6 #the output will be
7 e
8 d
9 u
10 r
11 e
12 k
13 a
```


Key elements in Python – 7.Functions

A function in Python is a block of code which will execute whenever it is called. We can pass parameters in the functions as well.

```
1 | def my_func():  
2 |     print('function created')  
3 |  
4 | #this is a function call  
5 | my_func()
```

Online contents that will help you learn programming languages that will be covered in this course

- Introduction to Python : <https://www.edureka.co/blog/introduction-to-python/>
- Youtube vidoes
 - Python (Programming language that will be mainly covered during the semester)
 - <https://www.youtube.com/watch?v=Y8Tko2YC5hA&list=LLoK27u6DBBrErwE0793NWhg&index=6&t=0s>
 - <https://www.youtube.com/watch?v=rfscVS0vtbw>
 - <https://www.youtube.com/playlist?list=PLFD32AF85033E6DDC>
 - R (Programming language that will be covered only partially during the semester)
 - <https://youtu.be/V8eKsto3Ug>
- Highly recommend to visit and learn from the **code academy**, which is a free website that provides high quality and interactive learning courses of programming languages
 - <https://www.codecademy.com>

Online contents that will help you learn programming languages that will be covered in this course

The screenshot shows the Codecademy homepage. At the top, there's a navigation bar with 'codecademy', 'Catalog', 'Resources', 'Community', 'Pricing', and 'For Teams'. A search bar and 'Log In'/'Sign Up' buttons are also present. A yellow banner at the top right says 'Not sure where to begin? Take our quiz →'. Below this is the 'Explore the catalog' section with 'Trending subjects & languages' and three cards for 'AI', 'Python', and 'JavaScript'. A 'RECENTLY ADDED' section features 'Explore new AI courses' with a pyramid icon. A sidebar on the left lists various programming languages and subjects.

The screenshot shows the 'Learn Python 3' course page. The course title is 'Learn Python 3' with a rating of 4.6 stars and 5,621 ratings. A 'Start' button is visible, along with a note that 2,180,921 learners are enrolled. Below the course title, there are four key metrics: Skill level (Beginner), Time to complete (Approx. 25 hours), Certificate of completion (Included with paid plans), and Prerequisites (None). The 'About this course' section describes Python as a flexible and easy-to-read language. The 'Skills you'll gain' section lists three skills: writing Python 3 programs, simplifying code, and taking skills off-platform. A 'Syllabus' section shows 14 lessons, 12 projects, and 12 quizzes, with an 'Expand all sections' link.

The screenshot shows the 'Python' catalog page. The title is 'Python' and it describes Python as a general-purpose, versatile, and powerful programming language. A 'Related resources' box lists links for 'Docs: Python', 'Cheatsheets', 'Articles', 'Community Forums', and 'Projects'. A 'New to Python? Start here' section features a 'Learn Python 3' course card with a 'With Certificate' badge, 'Beginner Friendly' label, and '14 Lessons' count. A 'Blog Post' titled 'What is Python used for?' is dated 2 November 2022.

Using terminal program, Xshell, to use remote server

- It's ssh (Secure Shell Protocol) client that enables users from remote place to connect to host server through secured way
- It needs either private public key
- You can use Xshell7, one of ssh client that is free of charge, to connect to our server in BIGLAB (CentOS)

Get access for downloading the software

(<https://www.netsarang.com/ko/xshell-download/>)



XMANAGER

XSHELL

XFTP

XLPD



다운로드

개요

구매하기

라이선스 종류를 선택하세요

기재하신 정보 중 이름, 회사(기관), 이메일 정보는 제품 출시나, 업데이트 정보 등을 신속하게 제공하기 위해 사용됩니다. 이 정보는 서비스 기간 동안 보관하고 있으며, 이외의 다른 목적으로 사용되지 않습니다.



정품 사용자

제품 키(필수)

버전을 선택하세요(필수)

버전 7 버전 6 버전 5

다운로드



30일 평가판

이름(필수)

이메일(필수)

다운로드 링크가 이메일로 보내집니다.

회사

향후 제품 안내나 프로모션 등과 관련한 이메일 수신에 동의합니다.

평가 시작

가정 및 학교 내 사용자를 위한 무료 라이선스

> 무료 라이선스 페이지

관련 다운로드

- > 소프트웨어 사용권 및 개인 정보 이용 동의서
- > 사용자 매뉴얼
- > 데이터시트

Get access for downloading the software

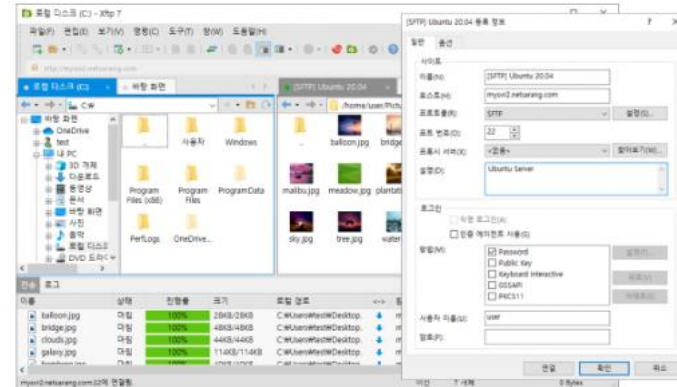
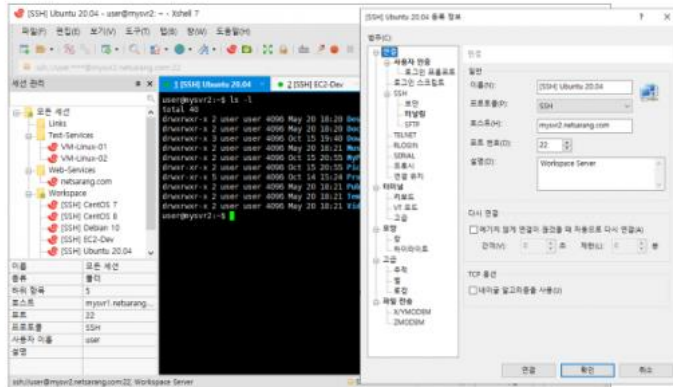
- Download both Xshell & Xftp



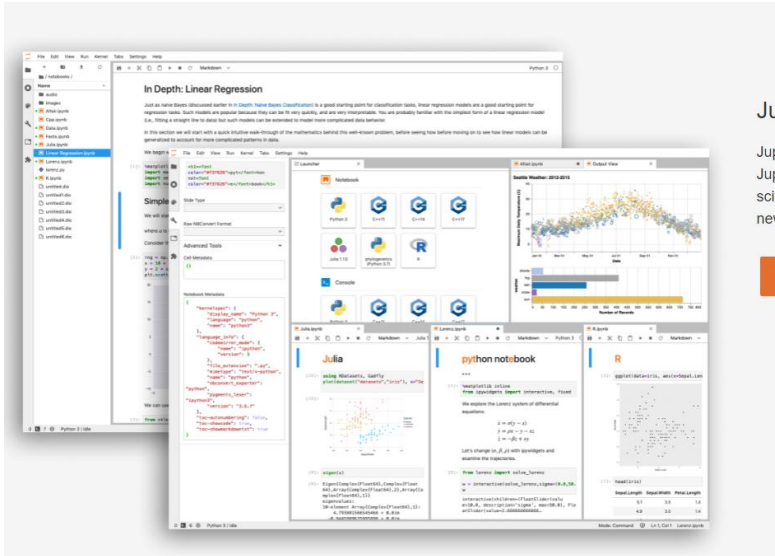
업계 최고의 강력한 SSH 클라이언트



네트워크를 통한 수고스럽지 않은 파일 전송



Jupyter notebook

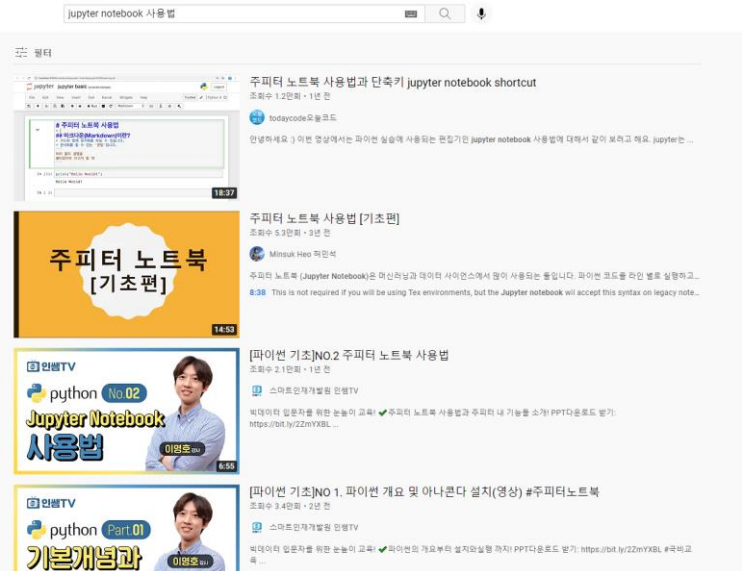


JupyterLab: Jupyter's Next-Generation Notebook Interface

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

Try it in your browser

Install JupyterLab




생물정보학 실습 환경

Xshell

```
BIGLAB_KT - Xshell 5 (Free for Home/School)
[파일] [편집] [보기(N)] [도구(T)] [탐색] [찾기(F)] [도움말(H)]
100.104.118.161.22
BIGLAB_KT
1 BIGLAB_KT 2 BIGLAB_KT 3 BIGLAB_KT
-rw-rw-r-- 1 kyungtae kyungtae 8.0K Aug 27 10:18 getFoldAssociation.py
-rw-rw-r-- 1 kyungtae kyungtae 5.6K Aug 16 16:45 getGeneExp.py
-rw-rw-r-- 1 kyungtae kyungtae 1.3K Aug 16 16:45 getGeneSummary.py
-rw-rw-r-- 1 kyungtae kyungtae 3.2K Jul 5 03:15 getSigGeneSigAcSBed.py
-rw-rw-r-- 1 kyungtae kyungtae 8.6K Jul 7 17:58 getTfGeneInteract2.py
lrwxrwxrwx 1 kyungtae kyungtae 31 Jun 13 15:26 getTfGeneInteract.py -> ../diffBind/getTfGeneInteract.py
-rw-rw-r-- 1 kyungtae kyungtae 8.2K Jun 13 15:14 getTfGeneInteract.py.bak
-rw-rw-r-- 1 kyungtae kyungtae 7.9K Jul 16 16:40 getTfGeneInteract.pyc
lrwxrwxrwx 1 kyungtae kyungtae 35 Dec 23 2020 getModule.py -> /home/kyungtae/mainSrc/getModule.py
-rw-rw-r-- 1 kyungtae kyungtae 4.7K Jun 13 15:57 getModule.pyc
-rw-rw-r-- 1 kyungtae kyungtae 4.7K Aug 16 16:03 makeGenomicRnaRegion.py
-rw-rw-r-- 1 kyungtae kyungtae 2.7K Aug 25 16:47 makeIscBed.py
-rw-rw-r-- 1 kyungtae kyungtae 5.5K Jun 13 15:15 parseDiffTF.py
-rw-rw-r-- 1 kyungtae kyungtae 6.2K May 28 10:46 parseDiffTF.pyc
-rw-rw-r-- 1 kyungtae kyungtae 15K Jun 14 15:28 parseHomeResult.py
-rw-rw-r-- 1 kyungtae kyungtae 15K Jul 16 16:40 parseHomeResult.pyc
lrwxrwxrwx 1 kyungtae kyungtae 46 Dec 23 2020 programExecuteModule.py -> /home/kyungtae/mainSrc/programExecuteModule.py
-rw-rw-r-- 1 kyungtae kyungtae 3.5K Jul 16 16:40 programExecuteModule.pyc
dwxr-xr-x 5 kyungtae kyungtae 4.0K Aug 25 14:59 rsrc
dwxr-xr-x 2 kyungtae kyungtae 4.0K Jul 6 13:37 Tobias
[kyungtae@biglab-master atac_src]$ cd diffBind/
[kyungtae@biglab-master diffBind]$ ll
total 92K
-rw-rw-r-- 1 kyungtae kyungtae 8.0K Jul 16 00:53 diffBind.R
-rw-rw-r-- 1 kyungtae kyungtae 1.5K Jul 26 14:27 diffBindToBed.py
lrwxrwxrwx 1 kyungtae kyungtae 37 Jun 7 15:18 fastaModule.py -> /home/kyungtae/mainSrc/fastModule.py
-rw-rw-r-- 1 kyungtae kyungtae 2.1K Jun 15 10:11 fastaModule.pyc
-rw-rw-r-- 1 kyungtae kyungtae 2.9K Aug 30 11:53 filterDiffBind.py
-rw-rw-r-- 1 kyungtae kyungtae 2.5K Jul 5 03:06 getIscSequence.py
-rw-rw-r-- 1 kyungtae kyungtae 7.8K Aug 26 13:50 getGeneL2fcByAcS.py
-rw-rw-r-- 1 kyungtae kyungtae 7.2K Aug 19 22:12 getGeneL2fcByAcS.pyc
-rw-rw-r-- 1 kyungtae kyungtae 5.0K Aug 24 17:02 getGeneMedAcS_heatmapInput.py
-rw-rw-r-- 1 kyungtae kyungtae 4.5K Aug 4 12:13 getHdacChip_gene_acs_l2fc.py
-rw-rw-r-- 1 kyungtae kyungtae 4.6K Jul 16 01:42 getMeanCountScatter.py
-rw-rw-r-- 1 kyungtae kyungtae 7.21 Jun 15 10:04 getNonDiffPeakBed.py
-rw-rw-r-- 1 kyungtae kyungtae 4.8K Aug 27 17:43 getPeakAnnoDistribution.py
-rw-rw-r-- 1 kyungtae kyungtae 7.8K Jul 14 16:20 getTfGeneInteract.py
-rw-rw-r-- 1 kyungtae kyungtae 8.1K Jul 13 13:35 getTfGeneInteract.pyc
[kyungtae@biglab-master diffBind]$ vi filterDiffBind.py
[kyungtae@biglab-master diffBind]$ vi getPeakAnnoDistribution.py
[kyungtae@biglab-master diffBind]$ vi getGeneL2fcByAcS.py
[kyungtae@biglab-master diffBind]$

def writeGeneL2fcDensity (gene_outputD, peak_outputD, gene_peak_interD, outputname):
    with open(outputname, "w") as outputopen:
        headers = "gene_l2fc\tpeak_group\n"
        outputopen.write(headers)
        for i_gene in gene_peak_interD.keys():
            try:
                gene_l2fc= gene_outputD[i_gene][0]
            except KeyError:
                continue
            peakL= gene_peak_interD[i_gene]
            if len(peakL)==1:
                peak= peakL[0]
            else:
                peak_count= list()
                for i_peak in peakL:
                    try:
                        peakinfo= peak_outputD[i_peak]
                    except KeyError:
                        continue
                    peakgroup, peak_logmean, peak_fdr, peak_l2fc= peakinfo
                    peakinfo= [i_peak, peak_logmean]
                    peak_countL.append(peakinfo)
                if peak_countL==[]:
                    continue
                peak_countL= sorted(peak_countL, key=lambda t: float(t[1]), reverse=True)
                # If multiple peaks are assigned to single gene, peaks of the highest mean count will
                # be chosen
                peak= peak_countL[0][0]
            try:
                peakgroup= peak_outputD[peak][0]
            except KeyError:
                continue
            outputline= str(gene_l2fc)+ "\t"+ peakgroup+ "\n"
            outputopen.write(outputline)
def writeGeneAcSL2fcScatter (gene_outputD, peak_outputD, gene_peak_interD, outputname):
    with open(outputname, "w") as outputopen:
        header= "geneid\tgenesymbol\tgene_l2fc\tgene_fdr\tacs_l2fc\tacs_fdr\n"
        outputopen.write(header)
        for i_gene in gene_peak_interD.keys():
            try:
```

 jupyter Untitled Last Checkpoint: 23분 전 (unsaved changes)



File Edit View Insert Cell Kernel Widgets Help

Trusted

            Code 

In [2]: `print("Welcome to 2023 Bioinformatics")`

Welcome to 2023 Bioinformatics

Homework until next week ☺

- ∨ VPN 신청
- ∨ Jupyter notebook 사용법 숙지 (Youtube video 등을 활용)
- ∨ Optional
 - 학기가 진행되는 동안 Code academy, 첨부해준 youtube 영상을 개인적으로 공부하기