1장 파이썬 소개











주	해당 chapter	주제
1	1장	파이썬 소개, 개발 도구 설치
2	2장	변수와 자료형
3	3장	수식과 연산자
4	4장	조건문
5	5장	반복문
6	6장	함수
7	7장	리스트
8		중간평가 및 기말 프로젝트 제안서 발표
9	8장	튜플, 세트, 딕셔너리
10	9장	
11	10장	파일 입출력과 예외 처리 부분만 자율적으
12	11장	tkinter를 이용한 GUI프로그래밍
13	12장	클래스와 객체
14	13장	pygame을 이용한 게임 작성 🌔
15	14장, 15장	파이썬 라이브러리 소개 •
16		기말 평가 및 기말 프로젝트 발표







- 컴퓨터에 일을 시키려면 인간이 컴퓨터에게 자세한 명령어 (instruction)들의 리스트를 주어야 한다.
- 프로그램 (program): 컴퓨터가 수행할 명령어를 적어놓은 문서



프로그램 소스 코드



- 컴퓨터는 사람의 언어를 이해할 수 없다!
- '프로그래밍 언어'는 컴퓨터가 이해하는 언어이다.





 인간이 프로그래밍 언어를 배워서 프로그램을 작성하면 컴파일러(또 는 인터프리터)라고 하는 통역 소프트웨어가프로그램을 기계어로 바 꾸어준다.





프로그램은 '프로그래밍 언어'로 작성된다. 프로그램을 만드는 사람을 '프로그래머'라고 한다.





 많이 사용되는 언어들에는 '파이썬', 'Java', 'C', 'C++', 'Javascript' 들 이 있다.



The popularity of Programming Language (PYPL) Ranking 2020





 1991년에 귀도 반 로섬(Guido van Rossum)이 개발한 대화형 프로그 래밍 언어







- 생산성이 뛰어나다.
- 간결하면서도 효율적인 프로그램을 빠르게 작성

C 언어	파이썬
<pre>#include <stdio.h></stdio.h></pre>	
<pre>int main(void) { printf("Hello World! \n"); return 0; }</pre>	print("Hello World!")



인터프리터 언어: 파이썬 프로그래머는 자신이 작성한 명령문의 결 과를 즉시 볼 수 있기 때문에 초보 프로그래머한테는 아주 바람직



큄파일러 ∨s 인터프리터

• 컴파일러는 전체 코드를 전부 번역한 후에 실행한다.



 인터프리터는 코드 한줄 씩 기계어로 번역하여 실행한다. 이전에 실 행했던 내용을 항상 기억하고 있다.









• 인터프리터는 한 문장씩 번역하는 동시 번역과 같다.













- 라이브러리가 풍부
- 라이브러리 설치가 쉽다.





- 파이썬의 특징을 요약해보자.
- 프로그래밍 언어의 역할은 무엇인가?





http://www.python.org/













참고 사항: 파이썬의 메이저 버전

파이썬을 처음 학습할 때 우리가 결정해야 하는 중요한 문제가 있다. "어떤 버전의 파이썬을 써야 할 것인가?" 이다. 파이썬에는 2.x와 3.x 버전이 있다. 문제는 이 버전들이 전혀 호환되지 않는다는 점이다. 2.x 버전 프로그 램은 3.x 버전 인터프리터에서 실행되지 않는다. 따라서 개발자들은 어떤 버전을 사용해서 개발할 것인지를 미 리 결정하여야 한다! 일반적으로 이전에 개발된 많은 라이브러리들을 사용하려면 2.x 버전을 사용하는 것이 좋 다고 한다. 하지만 미래를 대비하는 파이썬 개발자라면 3.x 버전을 사용해야 할 것이다. 이 책에서는 3.x 버전을 사용한다.

참고 사항: 파이썬의 마이너 버전

파이썬의 버전은 계속하여 업그레이드되고 있다. 이 책을 집필할 때는 파이썬의 최신 버전이 3.10.0이었지만 여 러분이 1장을 읽을 때는 버전이 달라질 수 있다. 따라서 뒤에 붙는 숫자는 신경 안 써도 된다. 또 한 가지 만약 외부 라이브러리를 설치할 때, 버전에 맞는 라이브러리를 찾지 못한다는 오류가 발생하면 파이 썬의 버전을 낮추어야 한다. 예를 들어서 최신 버전은 3.10이지만 3.8을 설치해야 하는 경우도 종종 발생한다. 3.8도 파이썬 홈페이지에서 여전히 제공된다.





윈도우의 시작 버튼을 누르고 [모든 프로그 램]->[Python 3.10]->[IDLE(Python 3.10 64-bit)]을 클릭한다.







DLE Shell 3.10.0	— n ×
File Edit Shell Debug Options Window Help win32 Type "help", "copyright", "credits" or "license()" for more in print("Hello World!") Hello World! >>>	print("Hello World!")를 입력하고 엔터키를 누른다. 이 문장은 화면에 "Hello World!"를 출력하라는 의미이다.
	Ln: 5 Col: 0



- 한글도 출력될까? "안녕하세요?"를 화면에 출력하여 보자.
- "programming에 입문하신 것을 축하드립니다."를 출력하여 보자.





- 파이썬 프로그램은 여러 줄의 명령어로 이루어진다. 한 줄의 명령어 를 문장(statement)라고 부른다.
- 함수(function)는 특별한 작업을 수행하는 명령어들의 모임이다.

print("Hello World!") 문자열은 따옴표로 둘러싸인 문자들의 print() 함수는 문자열을 화면에 출력한다. 모임이다. 텍스트 데이터를 나타낸다.



 문자열(string): 큰따옴표("...")나 작은따옴표('...') 안에 들어 있는 텍 스트 데이터



• 반드시 따옴표가 있어야 한다.

	IDLE Shell 3.10.0					—		×
File	Edit Sh	nell Debug	Options	Window	Help			
>>>	print Synta	(Hello W axError:	/orld) invalid	syntax	. Perhaps you forgot a comma?			^
>>>								~
							Ln: 7	Col: 0



• 덧셈, 뺄셈, 곱셈, 나눗셈을 컴퓨터를 이용하여서 실행하여 보자.

1	DLE Sh	ell 3.1(0.0					×	
File	Edit	Shell	Debug	Options	Window	Help			
>>>	10-	-20							^
	30								
>>>									v
							Ln: 10	Col:	0

1	DLE Sh	ell 3.1	0.0						×
File	Edit	Shell	Debug	Options	Window	Help			
>>>	10- -10	20							^
>>>	10* 200	20							
>>>	10/ 0.5	20							
>>>									~
							L	.n: 18	Col: 0



• 반지름이 10cm인 피자의 면적을 계산해보자.





>>> "나의 " + "고양이 " 나의고양이 >>>









• 오류 메시지를 읽어본다.

1	<pre>iDLE Shell 3.10.0 ile Edit Shell Debug Options Window Help >> PRINT(10+20) Traceback (most recent call last): File "<pyshell#4>", line 1, in <module> PRINT(10+20) NameError: name 'PRINT' is not defined</module></pyshell#4></pre>						
File	Edit Shell Debug Options Window Help						
>>>	PRINT(10+20) Traceback (most recent call last): File " <pyshell#4>", line 1, in <module></module></pyshell#4>			^			
>>>	PRINT(10+20) NameError: name 'PRINT' is not defined			*			
			Ln: 16	Col: 0			

6	DLE Shell 3.10.0	1. <u>11.</u> 1		×
File	Edit Shell Debug Options Window Help			
>>>	print(Good Bye)			^
	SyntaxError: invalid syntax. Perhaps you forgot a comma?			
>>>				~
			Ln: 29	Col: 0



- "생일축하!!"를 10번 출력하는 명령문을 만들어보자. 문자열 반복을 사용한다.
- 다음과 같은 명령문을 실행하면 오류가 발생한다. 원인을 알아보자.
 >> print("Hello)





- 대화형 모드(interactive mode): 콘솔에서 문장을 한 줄씩 입력하여 실행한다.
- 스크립트 모드(script mode): 파일을 만들어서 저장한 후에 파이썬 인 터프리터가 이 파일을 읽어서 파일 안의 문장들을 순차적으로 전부 실행한다.





• 코드가 복잡해지면 대화형 모드는 번거롭다.





def add5(x):
return x+5
<pre>def dotwrite(ast): nodename = getNodename() label=symbol.sym_name.get(int(ast[0]),ast[0]) print ' %s [label=*%s' % (nodename, label), if isinstance(ast[1], str): if ast[1].strip():</pre>
print '= %s"];' % ast[1]
else:
print '=]'
else:
<pre>print '"];' children = [] for in n, childenumerate(ast[1:]): children.append(dotwrite(child)) print,' %s -> (' % nodename for in :namechildren print '%s' % name,</pre>



• IDLE의 [File] → [New File] 메뉴를 선택한다

















- 1. 스파이더는 두 가지 모드로 사용할 수 있다. 두 가지 모드에 대하여 설명해보자.
- 2. 파이썬으로 프로그램을 작성할 때 대문자와 소문자를 구분할까?
- 3. 파이썬 소스 파일의 확장자는 무엇인가?
- 4. 스크립트 모드에서 소스 파일을 실행하려면 어떻게 해야 하는가?
- 5. 다음 소스를 스파이더에 입력한 후에 d:\sources\temp.py로 저장한 다. 이 소스를 실행해보자.

print(2+3) print(2-3) print(2*3) print(2/3)







print("2+3=", 2+3) print("2-3=", 2-3) print("2*3=", 2*3) print("2/3=", 2/3)





print(안녕하세요?) Print(" 이번 코드에는 많은 오류가 있다네요") print(" 제가 다 고쳐 보겠습니다.)

File Edit Format Run Options Window Help

print(안녕하세요?) Print("이번 코드에는 많은 오류가 있다네요") print("제가 다 고쳐 보겠습니다









 터틀 그래픽은 화면에서 거북이를 이용하여서 그림을 그리는 기능이 다.





import turtle t = turtle.Turtle() t.shape("turtle")	# 터틀 모듈 포함 # 터틀 생성 # 터틀 모양을 거북이로 설정
t.forward(100) t.left(90) t.forward(50)	#100 픽셀만큼 전진 # 90도 회전
turtle.done()	# 터틀 그래픽 종료





import turtle	# 터틀 모듈 포함
t = turtle.Turtle()	# 터틀 생성
t.shape("turtle")	# 터틀 모양을 거북이로 설정





t.forward(100)

#100 픽셀만큼 전진



지금부터 거북이를 움직이면 캔버스에 그림이 그려 진다. forward(100)은 거북이를 앞으로 100만큼 이 동시키는 명령어이다. 거북이가 움직이면서 직선이 그려지는 것을 볼 수 있다. 여기서 100은 100픽셀 을 의미한다. 즉 거북이가 앞으로 100 픽셀(pixel)을 움직이면서 직선을 그리게 된다.



t.left(90) t.forward(50)	# 90도 회전	
turtle.done()	# 터틀 그래픽 종료	





- 1. 거북이를 움직여서 정삼각형을 그려보자. 회전하는 각도를 몇 도로 하여야 하는가?
- 2. 거북이를 움직여서 정사각형을 그려보자. 회전하는 각도를 몇 도로 하여야 하는가?
- 거북이를 움직여서 6각형을 그려보자. 회전하는 각도를 몇 도로 하 여야 하는가?





import turtle
t = turtle.Turtle()
???



import turtle
t = turtle.Turtle()

t.shape("turtle") t.forward(100) t.left(120) t.forward(100) t.left(120) t.forward(100)

turtle.done()





• IDLE에서 [Help]->[Turtle Demo]를 누르면 다음과 같은 터틀 그래픽 예제들이 등장한다. [Examples]에서 예제를 선택하고 화면 하단의 [START] 버튼을 눌러본다.

🦸 forest - a Python turtle g	raphics example	- 0	×
Examples Fontsize Help			
Examples Fontsize Help bytedesign chaos clock colormixer forest fractalcurves lindenmayer minimal_hanoi nim paint peace penrose planet_and_moon rosette round_dance sorting_animate tree two_canvases yinyang Ist = [] brs = [] for t, br; t. peme t. peme t. peme t. peme t. peme t. peme t. peme t. peme	<pre>hon3 hics-example-suite: forest.py of 3 breadth-first-trees in tree. s see tree.py breadth-first'-rewrite of tten by Erich Neuwirth. see ivie.ac.at/erich.neuwirth/ Turtle, colormode, tracer, randrange erf_counter as clock ge(-n,n+1) nchlist, angledist, sizedis e+symRandom(angledist), factor*1.01**symRandom(size for angle, sizefactor in istance, parts, angledist) (i.0 * distance)/parts) ze, level, widthfactor, bra e von turtles und Liste von rtle eine! anchlist in list(zip(tlist, b size(size * widthfactor) color(255 - (180 - 11 * lev 180 - 11 * level + sy 0) down() mfd(t, size, level, angledis 1 ngle, sizefactor in branchli .fet(angle)</pre>		*
runtime	e: 0.89 sec.	START STOP CLEAR	



• IDLE에서 [Help]->[Turtle Demo]를 누르면 다음과 같은 터틀 그래픽 예제들이 등장한다. [Examples]에서 예제를 선택하고 화면 하단의 [START] 버튼을 눌러본다.

🧳 clock - a Python turtle graphics example		- ³ 🔲	
Examples Fontsize Help			
<pre>% #!/wsr/bin/env python3 # -*- coding: cpl252 -*- turtle-example-suite: tdemo_clock.py Enhanced clock-program, showing date and time Press STOP to exit the program! """ from turtle import * from datetime import datetime def jump(distanz, winkel=0): penup() right(winkel) forward(distanz) left(winkel) pendown() def hand(laenge, spitze): fd(spitze/2.0) It(120) fd(spitze/2.0) It(120) fd(spitze/2.0) It(120) fd(spitze/2.0) It(120) fd(spitze/2.0) def make_hand_shape(name, laenge, spitze): reset() jump(-laenge*0.15) begin_poly() hand(laenge, spitze) end_poly() hand_form = get_poly() reset() pensize(7) for in range(60): demomentations = for the form = form</pre>	Sunday Oct. 31 2021		>
use mouse/keys or STOP	START STOP CLEA		



• 터틀 그래픽으로 다음과 같은 집을 그려보자.









pendown()



penup()







- 프로그램은 컴퓨터에 내리는 명령어로 이루어지는 작업 지시서이다.
- 프로그래밍 언어는 컴퓨터가 이해할 수 있는 언어이다.
- 다양한 종류의 프로그래밍 언어가 있고 파이썬도 프로그래밍 언어의 일종 이다.
- 파이썬으로 프로그램을 작성하기 위한 개발 환경인 IDLE는 https://www.python.org/에서 다운로드 받을 수 있다.
- 파이썬 쉘에서는 프롬프트 >>> 다음에 코드를 입력하고 엔터키를 누르면 코드가 실행된다.
- 산술 계산을 하는 파이썬 연산자에는 +, -, *, /가 있다.
- print()는 화면에 문자열이나 계산 결과를 출력할 때 사용하는 함수이다.
- 스크립트 모드를 사용하면 코드를 파일에 저장하였다가 한꺼번에 실행할 수 있다.









