|  |  |
| --- | --- |
| 교육 제목 | **개발기초역량교육** |
| 교육 일시 | 2021년 9월 16일 |
| 교육 장소 | 영우글로벌러닝 |
| **교육 내용** | |
| 오전 | 프로그래밍은 프로그램을 만드는 것인데, 프로그램은 미리 작성된 계획으로 누군가가 무엇을 했을 때 어떻게 할 것인지를 표시하는 것이고 컴퓨터 프로그램은 컴퓨터가 ‘무엇’과 ‘어떻게’를 하는 것이다. 한데 문제는 컴퓨터는 (컴퓨터를 어지간히 모르는 사람도 알듯이) 이진 숫자, 즉 이진법으로 작동되는데 문제는 어지간한 사람은 이진법으로 컴퓨터에게 직접 말하는 것이 매우 어렵다는 것으로, 이것을 해결하기 위해 중간에 사람이 이해하기 쉬운 언어(상대적이라는 것이 문제지만)를 만들었고 이것이 프로그래밍 언어이다. 그리고 그 언어로 만든 프로그램은 소스 코드로, 소스 코드를 해석하면 프로그램이 어떻게 작동하는지 알 수 있다. 필요한 설명을 하느라 파이썬 설명이 늦었는데, 파이썬은 앞의 이유로 만들어진 많은 프로그래밍 언어 중 하나이다. 많은 언어 중 파이썬을 쓰는 이유이자 장점은 배우기가 쉽고, 다양한 분야에서 활용되며, 대부분의 운영체제에서 동일하게 사용할 수 있다는 것이다. 다만 쉽다는 장점으로 인해 어쩔 수 없이 생긴 단점이 있는데, C언어에 비해 10~350배나 느리다는 것. 물론 요즘엔 컴퓨터 성능이 많이 좋아져서 어지간한 프로그램은 체감하기도 힘들지만 게임 등의 연산이 많은 무거운 프로그램에서는 확실히 느려져서 연산이 많은 프로그램은 파이썬 대신 C언어 등의 속도가 빠른 프로그램을 쓴다.  파이썬은 많이 쓰이는 프로그램이라 다운방법이 다양한데, 여기서는 아나콘다라는 프로그램으로 파이썬과 다른 파이썬의 보조프로그램들을 받았다.  파이썬의 가장 작은 단위는 표현식으로 키워드가 여기에 속하고, 표현식이 모인 것이 문장, 문장이 모인 것이 프로그램(주석과 식별자는 이정도가 되어야 나타난다.)이 된다.(앞서 말했던 그 프로그램 맞다.) 키워드는 개발자가 명령을 지정한 단어로, 명령어인지라 사용자가 임의로 짓는 이름인 식별자로 쓸 수 없기 때문이다. 식별자는 앞서 말했듯 사용자가 임의로 이름붙이는 단어인데 대문자로 시작하는 단어는 캐멀 케이스, 언더바(\_)가 들어간 단어는 스네이크 케이스라고 부른다.(이유는 글자와 동물의 모양에서 따온 것이다.) 또한 주석은 설명을 위해서 만든 것으로, 프로그램에는 영향을 주지 않고 프로그램을 보는 다른 사용자들을 위해 만든 설명이다. print()는 ()안에 글자를 입력하면 프로그램이 출력을 하게 해준다. 참고로 주피터 노트북에서 프로그램의 실행방법은 crtl+앤터를 눌러 실행하는 것과 shift+엔터를 눌러 실행하는 동시에 새 칸을 만드는 방법이 있다.  자료는 프로그램이 처리하기 위한 모든 것으로 사진이나 메시지, 게임의 정보 등을 가리킨다. 이것은 가장 기본적인 단위로 이걸 잘 알고나서 원하는 대로 써야 이것을 토대로 더 복잡한 프로그램을 만들 수 있고 내가 원했던 프로그램도 만들 수 있다. type()은 자료의 형태인 자료형을 알려주는 함수인데 결과값으로 str(string이란 뜻이다)은 문자열, int는 숫자를 뜻한다. 문자열은 print()안에 큰따옴표인 ””나 작은 따옴표인 ‘’사이에 문자열을 넣으면 문자를 출력하는데 문제는 같은 따옴표가 겹치면 오류를 일으켜 입력을 할 수 없다는 것이다. 이 때 큰 따옴표(작은 따옴표)안에 작은 따옴표(큰 따옴표)를 넣는 방식으로 서로 다른 따옴표를 넣어 해결하는 방법과 이스케이프 문자인 역슬래시나 원화표시(\)를 넣어 같은 따옴표(예: “\”문자열\””)로 해도 오류가 없게 하는 방법이 있다. 그리고 \n을 뒤에 붙이면 줄을 바꾸고, \t를 붙이면 탭으로 인식해 크게 띄우는데, 이게 여러 개면 줄바꿈을 했는지 알기 힘들어 일일이 \n을 확인해야 하고, 찾지 못해 실수할 확률도 올라가는데 파이썬은 같은 따옴표 세 번(“”” “””, ‘’’ ‘’’)을 입력하면 굳이 \n을 입력하지 않고 문자열에서 앤터만 눌러도 줄바꿈이 되어 훨씬 보기가 편하다. 다만 따옴표 세 번은 첫 번째 줄과 마지막 줄에 줄바꿈이 무조건 들어가므로 이것을 원치 않을 때는 \를 두 번 입력해(예시: ‘’’\문자열 줄바꿈 문자열\’’’)이런 현상이 일어나지 않게 할 수 있다. 또, 문자열에서 +는 연결 연산자 역할로 문자열을 연결해주고, \*는 반복 연산자로 앞이나 뒤에 붙은 숫자의 수만큼 문자열을 반복해 출력한다. |
| 오후 | 문자열에서 문자는 하나씩 한 칸에 넣는데, 이것의 위치를 인덱스라고 한다. 이것은 1부터 시작하는 원 인덱스와 0부터 시작하는 제로 인덱스가 있는데 파이썬은 제로 인덱스이다. 다만 특이한 건 -1부터 시작하는 음수인데, 이것은 0과 양수와는 반대로 오른쪽부터 -1로 시작해 동양의 세로쓰기 순서가 생각나게 한다. 문자열 범위 선택 연산자인 슬라이싱은 [:]으로 여기에 숫자를 입력하면 처음 표기한 숫자부터 나중 표시한 숫자까지 선택이 된다. (예: [1:3]) 다른 프로그래밍 언어는 마지막 숫자까지 포함하는 것도 있지만 파이썬은 끝 숫자 전 문자까지만 출력된다. 여기서 끝 숫자를 생략하면 (예: [1:]) 시작만 입력받은 인덱스부터시작해 끝까지 출력하고 처음 숫자를 생략하면(예:[:3])끝 문자만 정해지고 시작은 처음부터 출력한다. 여기서 len()함수는 글자수를 출력한다.  다음은 숫자를 입력하고 원하는 값을 출력할 차례인데 이를 위해 파이썬에선 정수와 실수가 나뉘고 정수는 소수점이 없고 실수(부동 소수점)는 소수점이 있는 숫자인 것을 알아야 한다. 그리고 숫자는 정수는 int, 실수는 float라는 키워드로 입력한다. 숫자연산자는 기본적인 사칙연산인 +, -, \*, / 와 소수점 아래는 버리고 출력하는 정수 나누기 연산자 //와 나누었을 때의 나머지 값만 표시하는 %, 제곱 연산자를 만들 수 있는 \*\* 가 있으며 연산 우선 순위는 수학에서 배운 것과 같이 곱셈과 나눗셈부터 순서대로 하고 괄호가 있으면 괄호먼저 계산한다.  변수는 값을 저장하는 식별자로 모든 자료형을 저장할 수 있다. 여기서 변수과 값은 변수=값(예: 상자=고양이집)의 순서로 되고 이것이 매우 중요한데, 이것만 보면 직관적으로 보이지만 실제로 만들다보면 처음 생각보다 직관적이지 않은 상황이 많이 나와 이 순서를 꼭, 정확히 기억해야 나중에 괴롭지 않다. 여기서 독특한 복합 대입 연산자가 나오는데, +=, -=, \*=, /=, %=, \*\*= 가 그것으로 계산 방법은 앞에 나온 연산자들의 계산방법과 같지만, a = a+10을 a += 10으로 표시한다는 것이 특이한 점이다. 이 다음에 사용자가 값을 입력할 수 있는 input()이 나오는데 여기서 값을 받아 프로그램이 계산을 하는 등의 작업을 한 뒤 출력하는 것이다. 문제는 input은 언제나 문자열로만 받고 숫자도 문자열로 바꾸기 때문에 문자열을 다시 바꿔주는 작업을 해야 하고 int()함수와 float()함수에 넣어주는 것으로 해결된다. 반대로 숫자를 문자열로 바꾸는 방법은 str()함수를 쓰면 된다. |