| 2차시 | 1 | 번 |  | 실습: ☑ | 과제 : □ | 평가 : □ |
| --- | --- | --- | --- | --- | --- | --- |
| 문제1) 컴퓨터는 숫자이든 문자이든 0과 1로만 이해한다. 왜 그런지 설명하시오.  **정답1) 컴퓨터의 하드웨어는 전기적 신호로 동작하는데, 디지털 회로는 두가지 상태, 켜짐(on)과 꺼짐(off) 상태로 구성된다. 그리고 컴퓨터은 이진법으로 데이터를 처리 하는데, 이진법은 두가지 숫자(0과 1)만을 사용하여 모든 데이터를 표현할 수 있는 체계이다. 이진법을 사용하는 이유는 회로 설계가 단순해지고, 신호가 단순해 오차 가 적어 신뢰성이 높아진다. 이렇게 전기적 특성과 수학적 측성이 결합되어 매우 신뢰할 수 있고 효율적인 데이터 처리를 가능하게 한다.**  문제2) 2 문자열에서 문자열을 빼면 오류가 발생한다. 어떤 오류인지 설명하시오.  덧셈 연산을 해도 동일한 오류가 발생하는지 설명하시오  **정답2) 다음에 보인 것처럼 문자열과 문자열 사이에 뺄셈 연산이 지원되지 않는다고 오류가 발생한다.**  **>>> "abc" - "ab"**  **Traceback (most recent call last):**  **File "<stdin>", line 1, in <module>**  **TypeError: unsupported operand type(s) for -: 'str' and 'str'**  **덧셈 연산은 다음에 보인 것처럼 제대로 동작한다. 문자열의 덧셈 연산은 문자열을 연결시키는 것으로 보인다.**  **>>> "abc" + "ab"**  **'abcab'**  문제3) 파이썬의 정수와 실수에 크기 제한이 있는지 없는지 답하시오  **정답3) 파이썬의 정수는 크기 제한이 따로 정해져 있지 않다. 컴퓨터의 메모리가 허용 하는 범위 내에서 숫자 표현이 가능하다. 실수는 크기 제한이 있다. 실수는 4.9 x 10^-324 ~ 1.8 x 10^308 범위 내의 값을 표현할 수 있다.**  문제4) 파이썬 코드로 '그가 안녕하세요라고 말했다'라는 문장을 출력한다고 가정하자. 출력하는 문자열에서 "안녕하세요"는 직접 말한 것으로 따옴표를 붙여야 한다면, 어떻게 코드로 작성할까?  **정답4) 직접 말한 것으로 인용 부호를 붙인다면 큰 따옴표를 사용할 가능성이 높다. 따라서 두 가지 방법으로 표현할수 있다. 첫 번째는 전체 문자열을 작은 따옴표로 묶는 것이다.**  **>>> print('그가 "안녕하세요"라고 말했다')**    **두 번째는 문자열을 큰 따옴표로 묶고 이스케이프 시퀀스를 이용해서 표현하는 것이다.**  **>>> print("그가 \"안녕하세요\"라고 말했다")**  문제5) 유니코드에 대한 설명 중 옳지 않은 것은?  (1) 유니코드는 ASCII 의 단점을 보완하기 위해 고안되었다.  (2) 유니코드는 세계에서 사용되는 많은 언어들을 몇개의 표를 이용해서 컴퓨터가 이해할 수 있는 숫자 형태로 상호 변환하기 위해 만들어졌다.  (3) 유니코드를 사용하면 코드를 수정하지 않고 다른 나라 언어를 표현할 수 있어, 다국적 프로그램을 작성하는데 용이해진다.  (4) 파이썬 3는 내부적으로 유니코드를 사용해서, 변수 이름이나 함수 이름들을  영어가 아닌 다른 나라 언어(한국어 포함)로 작성하는 것도 가능하다.  **정답5) 모두 옳음.** | | | | | | |

| 문제6) 전화번호 111-222-3333은 하이픈이 섞여 있는 문자열이다. 형변환을 하기 위해 int(111-222-3333)을실행하면 어떤 결과가 출력되는지 적으시오.    **정답6)**  **>>> int(111-222-3333)**  **-3444**  **만약 문자열로 취급하기 위해 따옴표를 붙이면 '-'는 정수가 아니므로 변환할 수 없어 오류가 발생한다.**  문제7) 문자열과 이스케이프 시퀀스를 이용해서 아래와 같이 따옴표를 출력하는  프로그램을 작성해 본다.  She said, "It's a beautiful day."  He replied, "Yes, it is!"  **정답7)**  **print('She said, "It\'s a beautiful day."')**  **print("He replied, \"Yes, it is!\"")**  문제8) 유니코드를 사용하여, 그리스 문자 Omega와 미소짓는 얼굴의 이모티콘을  출력해 보세요.  **정답8)**  **print("그리스 문자 오메가: \u03A9")**  **print("미소짓는 얼굴 이모티콘: \U0001F600")** |
| --- |

| 문제9) 리터럴 상수에 관한 설명 중 틀린것은?  (1) 코드에서 직접적으로 명시되는 숫자값이나 문자열이다.  **(2) 불변하는 파이값 3.14를 PI = 3.14 라는 코드로 PI 라는 변수에 할당했다.**  파이값은 변경되지 않으므로 PI는 리터럴로 간주된다.  (3) 리터럴의 사용은 코드의 가독성을 높이고 이해하기 쉽게 만든다.  (4) 일부 컴파일러는 리터럴값을 인식하여 코드를 최적화 할 수 있다.    문제10) 실수 124.578 를 소수점 두자리까지만 출력하는 코드를 작성하세요.  **a = 123.456**  **num = f"{a:.2f}"**  **print(num)**  문제11) 세줄로 나뉘어 있는 다음 문자열을 그대로 화면에 출력할 수 있도록 코드에 문자열 상수로 입력하고, 세번 화면에 출력해 본다.  <요구사항> 문자열에 대한 덧셈 또는 곱센 연산자 활용  디비딥 디비딥 딥디비딥  디비디비딥 딥디비딥  디비딥 디비딥 딥디비딥  **정답11)**  **print("디비딥 디비딥 딥디비딥\n디비디비딥 딥디비딥\n디비딥 디비딥 딥디비딥 \n" \* 3)**  문제12) 다음의 2진수와 8진수는 십진수로 얼마인가요?  (1) 2진수 0B1010, 0B1111 : **10, 15**  (2) 8진수 0o10, 0o17 : **8, 15**    문제13) 네 자릿수 정수를 입력받은 후 그 정수를 역순으로 출력하는 프로그램을 작성하세요.  **정답13)**  **temp = input("네자리 정수를 입력하세요: ")**  **reverse = temp[::-1]**  **print(reverse)** |
| --- |

| 2차시 | 14 | 번 |  | 실습: ☑ | 과제 : □ | 평가 : □ |
| --- | --- | --- | --- | --- | --- | --- |
| 문제14) 두 점 (1, 4)와 (3.5, 2.4) 사이의 거리를 구하는 프로그램을 작성하시오.  (루트인 제곱근은 지수승 1/2로 계산)  **정답14)**  **distance = ((3.5 - 1) \*\* 2 + (2.4 - 4) \*\* 2) \*\* (1 / 2)**  **print(f"두 점 사이의 거리: {distance}")**  **수학함수를 사용한다면**  **import math**  **distance = math.sqrt((3.5 - 1) \*\* 2 + (2.4 - 4) \*\* 2))**  문제15) 265일은 몇월 며칠인지 출력하는 프로그램을 작성하시오.  (한달은 30일로 계산)  **정답15)**  **days = 265**  **months = days // 30**  **days %= 30**  **print(f"{months}월 {days}일 입니다.")**  문제16) 2개의 실수를 입력받아 더하기, 빼기, 곱하기, 나누기의 연산을 출력한다.  이후 다시 하나의 연산식을 한 줄로 입력받아 그 결과를 출력하는 프로그램을 작성해 보자.  **정답16)**  **num1 = float(input('첫번째 수 입력 >> '))**  **num2 = float(input('두번째 수 입력 >> '))**  **print('합: ', num1 + num2)**  **print('차: ', num1 - num2)**  **print('곱하기: ', num1 \* num2)**  **print('나누기: ', num1 / num2)**  **expression = input('연산식 입력(예: 3.2 + 4 \* 1.5) >> ')**  **print('연산식: ', expression, '결과: ', eval(expression))**  문제17) 다음 코드의 오류에 대해 설명하시오.  (1) data = input()  print(data + 1)  **(1) input() 함수로 입력된 값는 문자열로 인식되기 때문에 직접 연산이 안된다.**  (2) num = int('3.14')  print(num – 1)  **(2) 3.14는 실수이므로 정수로 바로 변환할수 없다**  (3) str = '34'  print('결과 >> ' + int(str))  **(3) 파이선에서는 문자열과 정수를 직접 연결할수 없다.** | | | | | | |

| (4) a, b = 5, 7  a = b = 8 = c = 9  **이 코드에서 8 = c 부분이 문제를 일으킵니다. 파이썬에서는 왼쪽에 변수나 속성이 있어야 하며, 오른쪽에 값을 할당할 수 있습니다. 그러나 8은 정수 리터럴로 상수 값이며, 상수 값은 할당의 왼쪽에 올 수 없습니다. 따라서 8 = c와 같은 표현은 파이썬 문법에 맞지 않으므로 SyntaxError가 발생합니다.**  **구체적으로는 SyntaxError: can't assign to literal 오류가 발생합니다. 이 오류 메시지는 리터럴 값(여기서는 정수 8)에는 값을 할당할 수 없다는 의미입니다.**  문제18) 십진수의 한 숫자를 입력받아 2진수, 8진수, 16진수로 변환하는 프로그램을  작성하시오.  **정답18)**  **num = int(input('숫자를 입력하세요 >> '))**  **num\_binary = bin(num)**  **num\_octal = oct(num)**  **num\_hexadecimal = hex(num)**  **print(f"{num}의 2진수는 {num\_binary}, 8진수는 {num\_octal}, 16진수는 {num\_hexadecimal } 입니다.")**  문제19) 0.1 과 0.2를 더하여 그 결과를 출력하는 프로그램을 만들고, 이 연산의 결과가 기대하는 값과 다를 때 그 이유를 생각해 보시오.  **정답19)**  **r = 0.1 + 0.2**  **print(r)**  **결과는 0.300000000000004**  **파이선에서 부동 소수점의 정확한 표현이 제한되어 있기 때문입니다. 컴퓨터에서 소수를 이진수로 표현할 때, 일부 소수를 정확하게 표현할 수 없어서 근사값으로 표현됩니다. 이러한 근사값으로 인해 부동 소수점 연산에서 약간의 오차가 발생할 수 있습니다.**  ~~문제20) 파이썬 tkinter 라이브러리를 활용하여 간단한 GUI를 생성하는 코드를 구현해 보세요.~~  **~~정답20)~~**  **~~import tkinter as tk~~**  **~~win = tk.Tk() #Create instance~~**  **~~win.title("Python GUI") # Add a title~~**  **~~win.mainloop() #Start GUI~~** |
| --- |