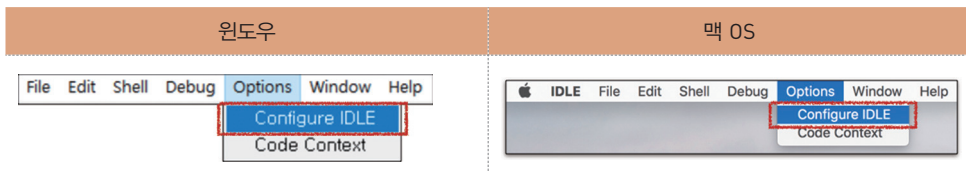


## 2

## IDLE 둘러보기



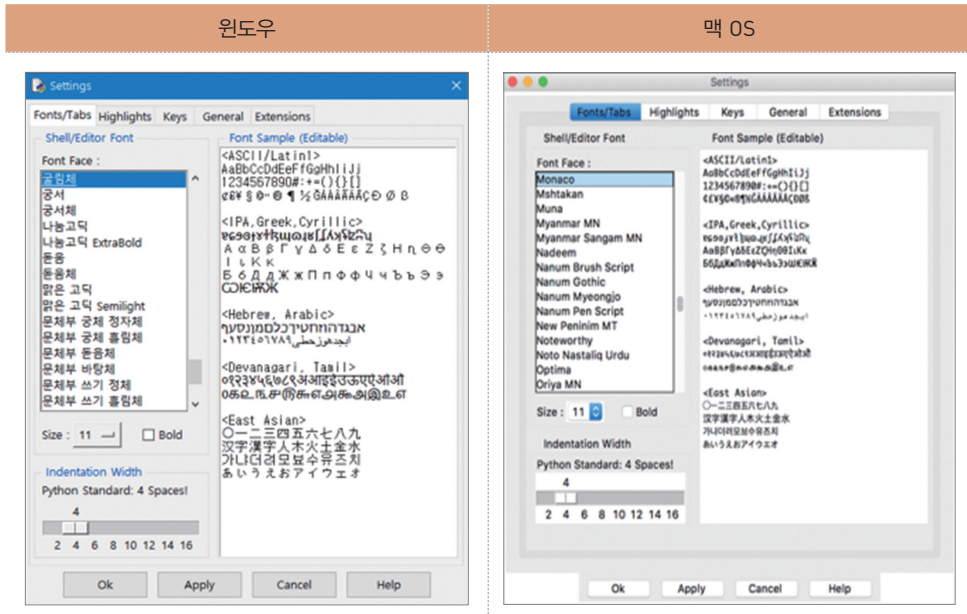
IDLE의 기본 기능들을 간략하게 둘러보자. IDLE 프로그램을 실행하고, IDLE이 제공하는 옵션 메뉴를 통해 환경설정을 각자의 기호에 맞게 개인화해보자. IDLE 창이 나타나면 그림 1과 같이 IDLE 상단 메뉴에서 **[Options] → [Configure IDLE]**를 선택하면 여러 가지 옵션을 변경할 수 있는 **[Settings]** 창이 열린다.



[그림 1] IDLE 설정 메뉴

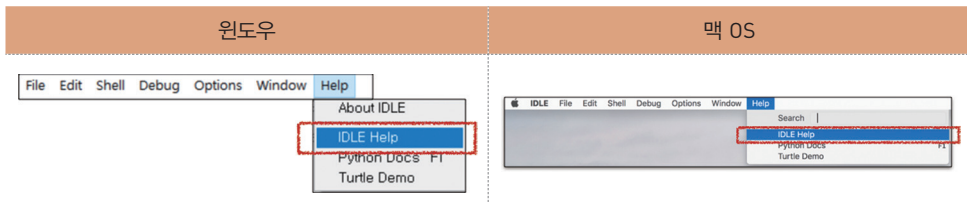
**[Settings]** 창에는 다섯 개의 탭이 있다. 이 중에서 가장 많이 사용하는 세 개의 탭에 대해 간단히 설명한다.

- **[Fonts/Tabs]** 탭에서는 글자 모양과 크기 및 탭 간격 등을 조절할 수 있다.
- **[Highlights]** 탭에서는 명령어와 기타 문자나 숫자, 주석 등을 종류에 따라 여러 가지 색 또는 글꼴로 다르게 구별하여 보여주는 구문 강조(syntax highlighting) 기능을 개인의 선호에 따라 다르게 세팅할 수 있다.
- **[General]** 탭에서는 파이썬 셸 창이나 프로그램 편집 창의 크기, 또는 편집과 관련된 일반적인 기능을 설정할 수 있다.



[그림 2] IDLE 세팅 창

IDLE이 제공하는 메뉴와 다양한 편집 기능 등을 좀 더 구체적으로 알고 싶으면 그림 3과 같이 IDLE의 도움말 기능인 [Help] → [IDLE Help]를 선택하면 된다. [IDLE Help] 창에서는 IDLE 프로그램이 제공하는 많은 기능들에 대한 설명을 살펴볼 수 있다.



[그림 3] IDLE 도움말 메뉴

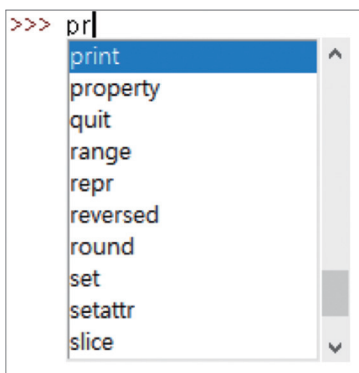
다음으로 IDLE이 제공하는 재미있는 그래픽 기반의 샘플 코드를 실행해 보자. 그림 4와 같이 IDLE 상단 메뉴에서 [Help] → [Turtle Demo]를 선택한다.



원한다면 [Examples] 메뉴에 나와 있는 여러 가지 샘플 코드를 실행해본다. [Python turtle-graphics examples] 창이 닫으면 데모 프로그램은 종료된다.

### 코드 실행해보기

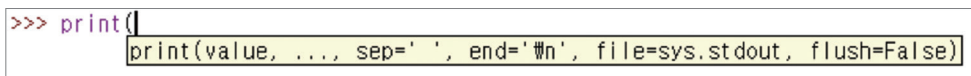
코드를 입력하기 전에 먼저 IDLE이 제공하는 기능 중에서 알아두면 유익한 명령어인 ‘자동 완성(auto-complete/show completions)’ 기능에 대해 알아보자. IDLE의 [파이썬 셸] 창에 나타난 프롬프트 >>> 우측에 깜박이는 커서에 ‘pr’이라고 입력한 후 **Tab** 키를 누르면 그림 7과 같은 화면이 나타난다.



[그림 7] IDLE 파이썬 셸의 명령어 자동 완성 기능

드롭다운 메뉴에서 ‘pr’로 시작하는 명령어부터 내림차순으로 보여준다. 여기서 print를 선택한다. 그러면 ‘print’라는 명령어가 자동으로 완성되는 것을 확인할 수 있다. 만약 ‘pri’이라고 입력한 후, **Tab** 키를 누르면 드롭다운 메뉴를 보여주지 않고 바로 ‘print’라는 문장을 완성한다. 이는 ‘pri’로 시작하는 파이썬 관련 용어가 print 하나 뿐이라서 드롭다운 메뉴를 사용자에서 보여줄 필요 없이 해당 명령어를 자동으로 완성하기 때문이다.

‘print’ 문장 바로 뒤에 띄어쓰기 없이 ‘(’를 입력하면 그림 8과 같은 화면이 나타난다.



[그림 8] IDLE 파이썬 셸의 명령어 매개변수 도움말

이처럼 파이썬은 개별 함수가 제공하는 모든 매개변수(파라미터, parameter) 옵션을 즉석에서 편리하게 참조할 수 있도록 도움말 기능을 제공하기 때문에, 복잡한 매개변수들을 일일이 다 기억하지 않아도 된다. 명령어 자동 완성 기능은 대화형 모드인 IDLE [파이썬 셸] 창과 나중에 설명할 인터프리터 모드인 IDLE [파이썬 프로그램] 편집 창, 둘 모두에서 제공되는 기능이다.

이제 [파이썬 셸] 창의 대화형 모드에서 명령어를 한 줄씩 입력한 후 실행해보자. 참고로 이 책 대부분의 예제 코드는 대화형 모드에서 입력하고 실행한 것이다. 코드를 한 줄 입력하고 그 실행 결과를 바로 확인할 수 있기 때문에 파이썬의 기능을 공부하기 매우 편리하다. 특히 파이썬 프로그래밍을 처음 배우기 시작하는 사람들에게는 매우 유용한 방식이다. 그러면 아주 간단한 코드를 입력해서 실행해 보자. 다음처럼 '2 + 9'를 입력한 후 **Enter** 또는 **return** 키를 눌러보자. 'print'는 소문자로 입력해야 한다.

```
>>> print(2 + 9)
11
>>>
```

2와 9의 합인 11이라는 결과를 출력한 후 다음 명령어를 기다리는 프롬프트가 다시 나타났다.

다시 나타난 프롬프트의 오른쪽에 다음 코드를 입력해보자. 이번 코드는 한 줄이 아니라 두 줄로 되어 있다.

```
>>> for i in range(10):
    print(i)
```

여기서는 단지 IDLE 대화형 모드를 체험하는 것이 목적이기 때문에 우리가 입력한 코드가 무엇을 뜻하는지 아직은 몰라도 된다. 위처럼 첫 줄을 작성하고 **Enter** 또는 **return** 키를 누르면 커서가 자동으로 아래 줄로 이동해서 네 칸 들여쓰기를 한 위치에서 커서가 깜빡일 것이다. 여기에서 둘째 줄의 코드를 입력한다. 파이썬에서는 들여쓰기로 명령문 블록을 구분한다. 이 코드의 경우 둘째 줄의 print문은 첫째 줄의 for

문에 속해 있는 명령문 블록이기 때문에 네 칸 들여쓰기를 한 것이다. IDLE의 파이썬 셸 대화형 모드에서는 첫 줄을 작성한 후 `Enter` 또는 `return` 키를 누르면 IDLE이 자동으로 네 칸 들여쓰기를 해준다. 들여쓰기에 관해서는 나중에 다시 상세히 설명하기로 한다. 둘째 줄 코드를 입력한 후 `Enter` 또는 `return` 키를 두 번 반복해서 누르면 코드가 실행된다. 대화형 모드에서 두 줄 이상의 명령문 블록의 경우에는 `Enter` 또는 `return` 키를 연속해서 두 번 눌러야 명령문 블록이 끝났음을 알리는 동시에 그 블록을 실행하게 된다. 만약 실행 도중 오류가 발생하면 코드를 다시 입력하면 된다.

코드를 정확히 따라했다면 다음과 같은 결과가 출력될 것이다. 만약 실수를 했다면 IDLE 창을 닫고 다시 열어서 다시 한번 시도해보자.

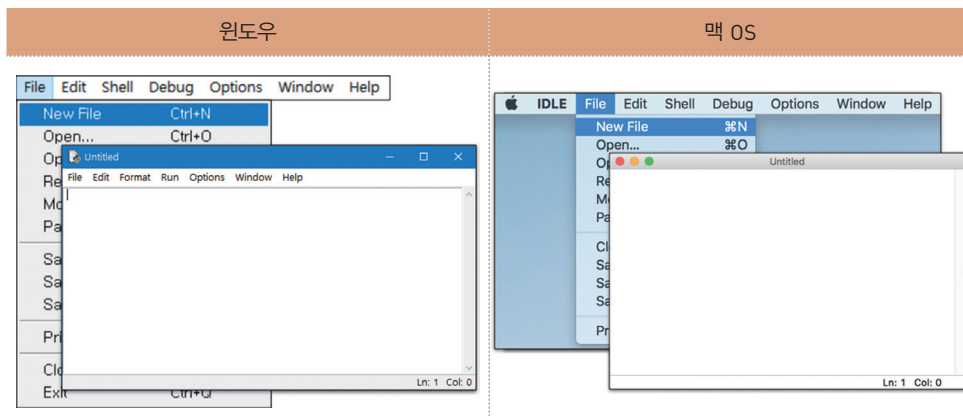
```
>>> print(2 + 9)
11
>>> for i in range(10):
    print(i)

0
1
2
3
4
5
6
7
8
9
>>>
```

지금까지 대화형 모드에서 간단한 파이썬 코드를 실행해 보았다. 이처럼 대화형 모드에서 코드를 실행하면 결과를 바로 확인할 수 있기 때문에 코드를 쉽게 테스트해 볼 수도 있고 계산기처럼 사용할 수도 있다. 그런데 만약 이러한 코드를 나중에 다시 실행해보려면 어떻게 해야할까? 위의 코드를 저장하지 않고 IDLE을 종료하면 지금까지 작성한 모든 코드를 잃어버리게 된다.

따라서 이번에는 방금 실행한 코드를 파일로 저장해보도록 하자. IDLE 상단 메뉴에서 **[File] → [Save]**을 선택해서 이름을 입력하면 파일이 저장되며, **[File] → [Open]**을 통해서 다시 파일을 열어보면 위의 화면에 나타난 모든 내용이 저장되어 있는 것을 확인할 수 있다. 하지만 이 파일은 앞의 내용을 그대로 저장했기 때문에 파이썬 명령어뿐만 아니라 파이썬 셸이 출력한 내용까지 모두 포함하고 있어서, 그대로 실행할 경우 오류가 발생하기 때문에 그렇게 유용하지 않다. 우리가 저장하고자 하는 것은 파이썬 명령문, 즉 파이썬 프로그램이다. 파이썬 프로그램을 파일로 저장하고 필요할 때 다시 그 파일을 열어서 실행할 수 있다면 매번 전체 프로그램을 다시 입력하는 수고를 하지 않아도 된다.

이제부터 파이썬 프로그램을 작성해서 저장한 후 실행하는 인터프리터 모드에 관해서 알아보자. 그림 9처럼 IDLE 창 상단 메뉴에서 **[File] → [New File]**을 클릭하면 **[Untitled]**란 제목의 빈 창이 나타난다. 이 창은 IDLE의 파이썬 코드 편집기이며 **[파이썬 프로그램]** 창이라 부른다. 파이썬 셸 창과는 달리 >>> 프롬프트가 보이지 않는다.



[그림 9] IDLE에서 새 파일 생성하기

파이썬 셸 창에서 앞서 작성한 명령문을 복사(copy)와 붙여넣기(paste)를 해서 여기에 입력해보자. 주의할 점은 복사해서 붙여넣기를 할 때, 대화형 모드의 표시인 >>> 프롬프트와 출력 결과는 제외시켜야 한다는 것이다. 이것들은 실제 파이썬 명령문이 아니다. 작성 결과는 다음과 같아야 한다.

```
print(2 + 9)
for i in range(10):
    print(i)
```

이제 이 파일을 IDLE 메뉴에서 **[File] → [Save]**를 선택해서 원하는 파일 이름을 입력하고 저장하자. 저장을 하지 않으면 나중에 다시 불러내서 사용할 수가 없다. 하지만 다행히 만약 저장을 하지 않고 IDLE 프로그램을 종료할 경우, IDLE이 친절하게 파일을 저장하겠는지 물어본다. 이때 만약 관리자 권한이 없어 사용자(로그인 ID) 폴더에 저장하겠냐고 물어보면 사용자 폴더에 저장한 후 이 폴더의 위치를 꼭 기억하도록 하자.

파일을 저장한 후 프로그램을 실행해보자. IDLE 창 상단 메뉴에서 **[Run] → [Run Module]** (또는 단축 키 **F5**)을 선택하면 프로그램이 실행된다. 프로그램이 실행되면 다음과 같은 화면이 나온다.

```
11
0
1
2
3
4
5
6
7
8
9
>>>
```

여기서 한 가지 주의할 점은 파이썬 프로그램 파일을 열어서 실행할 때 .py 확장자로 끝나는 파이썬 프로그램 파일을 직접 마우스로 더블 클릭해서 열기보다는 가급적 IDLE 프로그램에서 **[File] → [Open]** 메뉴를 선택해서 여러분이 저장한 파이썬 파일을 연 후에 **[Run] → [Run Module]** (또는 키보드의 **F5**)을 선택해서 프로그램을 실행하기를 권장한다. 파일을 더블 클릭한다고 해서 큰 문제가 발생하는 것은 아니지만, 시스템 세팅에 따라 예상치 못한 결과들이 발생할 수 있기 때문에 IDLE을 통해 파이썬을 열고 실행하기를 권장한다.



지금까지 IDLE의 대화형 모드와 인터프리터 모드에서 파이썬 프로그램을 작성해서 실행해 봤다. 그럼 지금까지 살펴 본 파이썬의 두 가지 모드를 간단히 정리해 보자.

대화형 모드(interactive mode)는 대화 창인 파이썬 셸에서 실시간으로 명령어를 입력하는 모드다. 대화형 모드를 실행하면 프롬프트(>>>))가 나타나는데 프롬프트는 사용자로부터 명령어를 입력받을 준비가 되었다는 것을 뜻한다. 그리고 명령어가 처리되면 다음 명령어를 기다리는 프롬프트가 다시 나타난다. 단, 잘못된 명령어를 입력해서 실행할 경우 오류 메시지가 출력된다. 그런 경우, 디버깅을 통해 오류를 수정해야 한다. 디버깅에 대해서는 다음 절에서 살펴보기로 하자. 대화형 모드는 주로 파이썬의 다양한 기능을 테스트할 때 편리하게 사용할 수 있으며 책에 있는 대부분의 예시 코드는 대화형 모드에서 작성되었다.

인터프리터 모드(interpreter mode)는 파이썬 프로그램 파일을 실행하는 모드다. 인터프리터 모드를 실행하기 위해서는 먼저 IDLE이나 다른 개발 툴의 텍스트 편집기를 사용하여 코딩하고 문서 형태로 저장해서 프로그램 파일을 생성해야 한다. 이때 파일 확장자를 .py로 설정하면 된다. 대화형 모드에서는 입력한 명령어들이 저장되지 않지만, 인터프리터 모드는 먼저 프로그램 파일을 저장한 후 파일 전체를 한번에 실행하는 모드다. IDLE의 경우 프로그램 파일을 실행하기 위해 메뉴에서 **[Run] → [Run Module]** 또는 **F5** 키를 선택하면 된다.

## 디버깅해보기

마지막으로 IDLE의 디버깅 기능을 살펴보자. 디버깅(debugging)이란 컴퓨터 프로그램 구문의 문법적 또는 논리적인 오류인 버그(bug)를 찾아내기 위해 테스트하고 수정하는 과정을 말한다. 만약 IDLE 대화형 모드에서 잘못된 명령어를 입력했을 경우 오류 메시지가 발생한다. 그러면 일부러 오류를 발생시켜 보도록 하자. **[파이썬 셸]**을 실행하여 새로 시작한 >>> 프롬프트에서 다음 내용을 정확히 그대로 입력하고 **Enter** 또는 **return** 키를 누른다.

```
>>> print('안녕 파이썬')
```

앞의 코드는 의도적으로 오류를 발생시키고자 문자의 마지막 부분 작은 따옴표(')를 생략하였다. 문자는 시작과 끝을 따옴표로 감싸야 한다. 즉, 작은 따옴표로 시작했으면 작은 따옴표를 끝에 붙여야 하며 큰 따옴표(")로 시작했으면 큰 따옴표로 끝을 감싸야 한다. 따라서 앞의 코드를 그대로 실행하면 그림 10과 같이 오류 메시지가 출력된다.

```
>>> print('안녕 파이썬')
SyntaxError: EOL while scanning string literal
>>>
```

[그림 10] IDLE 대화형 모드에서 파이썬 코드의 오류 화면

화면에는 SyntaxError 오류 메시지 블록이 붉은 색으로 표시가 되어 나타나기 때문에 어디서 오류가 발생했는지 금방 확인할 수 있다. 코드를 다음과 같이 수정해서 다시 실행해보자.

```
>>> print('안녕 파이썬')
```

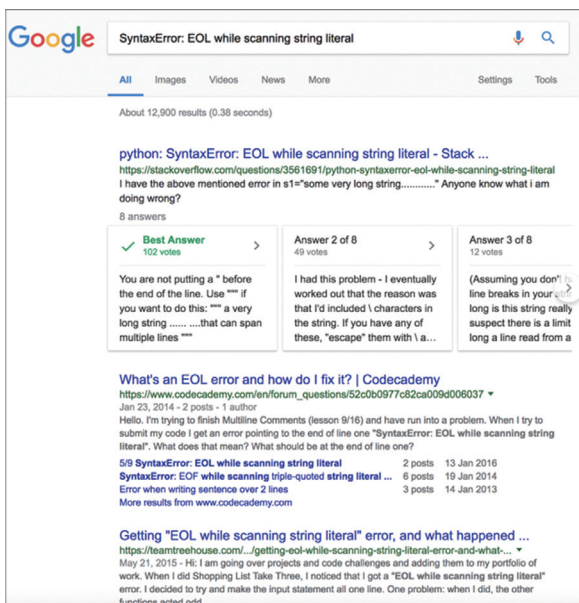
디버깅한 결과, 이제는 다음과 같이 아무런 오류 메시지 없이 코드가 잘 실행되는 것을 확인할 수 있다.

```
>>> print('안녕 파이썬')
안녕 파이썬
```

만약 오류 메시지의 의미를 알지 못하여 디버깅을 할 수 없다면 네이버나 구글 검색창에 오류 메시지를 그대로 복사한 후 붙여넣고 검색하면, 그림 11, 12에서 보는 것처럼 그 오류 메시지와 관련된 도움말이나 해결책을 찾을 수가 있다.



[그림 11] 네이버에서 오류 메시지 검색

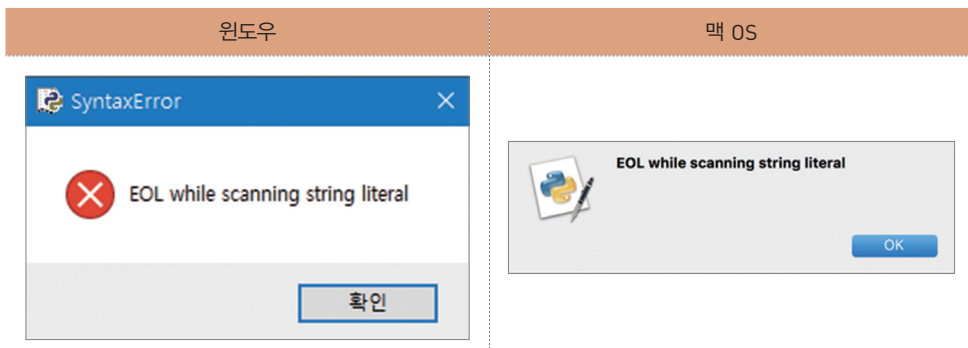


[그림 12] 구글에서 오류 메시지 검색

IDLE 인터프리터 모드에서도 디버깅은 매우 유사하다. IDLE 상단에 있는 메뉴에서 **[File] → [New File]**을 선택해서 새 **[파이썬 프로그램]** 창을 열고 오류가 있는 다음 코드를 입력한 후 저장한다.

```
print('안녕 파이썬')
```

저장한 뒤 코드를 실행하면 그림 13처럼 `SyntaxError` 오류 메시지 창이 나타나고 오류가 발생한 코드에는 붉은색 블록이 표시되어 나타난다. 앞에서 했던 것과 마찬가지로 디버깅하여 실행하면 정상적으로 코드가 실행될 것이다.



[그림 13] IDLE 인터프리터 모드에서 파이썬 코드의 오류 화면