|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | 🞂CLOSURE  5주차 업데이트 파일 |

|  |  |
| --- | --- |
|  | Tkinter 콜백함수에서는 인수를 전달해서 함수를 호출할 수 없다. 일반적으로 인수를 전달하면 콜백함수가 아니라 그냥 함수처럼 작동이 되서 액션을 취하지도 않았는데 동작되고 만다.  그렇다면 일반적으로 인수를 전달하면 안되겠다. 이게 뭔 소리냐면,  버튼에 있는 값을 스크롤드 텍스트 내부로 옮길 때 , insert메소드를 이용하는 함수에 인수를 전달하는게 일반적이겠지만 이게 안 된다는 것이다.  그래서 필자는 ‘클로저’라는 개념을 이용하였다.  클로저는 내부함수가 외부함수의 맥락을 접근할 수 있게 하는 것이다. 즉 클로저를 이용하면 내부함수가 외부함수의 변수, 매개변수 따위를 아무 조건없이 이용할 수 있다.  클로저의 예시는 아래와 같다.  Def Outer(a):  Def inner():  Print(a)  Return inner  A=1  B=Outer(A)  B()  A에 1을 저장하고 outer함수에 a인수 값을 전달한다. 이후 inner함수 자체를 리턴한다. B값은 그래서 inner가 어떤 함수인지에 대한 데이터가 된다. 만약 이 데이터를 가지고 함수를 호출하듯이 소괄호를 붙이고 컴파일 시, inner내부 함수가 호출되게 된다. 이 내부함수는 이 때 A변수를 가지게 되서 1을 출력할 수가 있다. 이게 일반적인 클로저 만들기이다.  쉽게 얘기해서 내부 함수를 만들고 외부 함수에서 내부 함수 자체를 리턴하면 된다.    그렇다면 왜 클로저를 사용하여만 할까? 그 이유는 command=(함수 호출)에 의해 리턴될 시,  Command=(내부 함수에 대한 데이터)가 되기 때문이다. 그럼 command속성에 의해 액션 시 , 내부 함수를 호출하는 일이 발생된다. 즉 내부 함수가 콜백 함수가 되는 것이고, 내부 함수는 외부함수가 가지는 매개변수를 가지게 되니 결국 콜백함수에 인수를 전달하는게 가능해지는 것이다.  위 소스는 falsetext라는 외부함수를 미리 만들고 내부 함수에 여러 리스트를 생성한 모습이다.  이 리스트들은 버튼이 눌림과 동시에 생성되고, 콜백함수로 보낸 값(n)에 따라서 (insert메소드를 이용하는 함수)에 리스트 요소를 전달할 수 있게 된다.  예를 들어 ‘9’텍스트 버튼을 클릭하면 text9에 9가 저장되고 리스트 요소로 설정한다.  이 text9값은 인덱스가 9인 리스트 요소를 보내는 방식으로 (insert메소드를 이용하는 함수)에 전달하여 스크롤드 텍스트 내부로 9가 출력되게 하는 것이다. |