

[소프트웨어프로젝트 II]

[PyQt 를 사용한 계산기_1]

20171661_이다운_class04

이시윤 교수님

1. 숫자 버튼 생성

34	32	# Digit Buttons
35	-	self.digitButton = [x for x in range(0, 10)]
36	-	
37	-	# self.digitButton[0] = Button('0', self.buttonClicked)
38	-	# self.digitButton[1] = Button('1', self.buttonClicked)
39	-	# self.digitButton[2] = Button('2', self.buttonClicked)
40	-	# self.digitButton[3] = Button('3', self.buttonClicked)
41	-	# self.digitButton[4] = Button('4', self.buttonClicked)
42	-	# self.digitButton[5] = Button('5', self.buttonClicked)
43	-	# self.digitButton[6] = Button('6', self.buttonClicked)
44	-	# self.digitButton[7] = Button('7', self.buttonClicked)
45	-	# self.digitButton[8] = Button('8', self.buttonClicked)
46	-	# self.digitButton[9] = Button('9', self.buttonClicked)
47	-	
48	-	for i in range(10):
33	+	self.digitButton = [x for x in range(0, calcNumber + 1)]
34	+	
35	+	for i in range(calcNumber + 1):
49	36	self.digitButton[i] = Button(str(i), self.buttonClicked)

- 규칙이 있는 반복되는 코드를 반복문으로 간결하게 만듭니다.
- 규칙을 가지고 변하는 부분이 숫자인 문자열이기 때문에 for 문으로 반복문을 작성한 후, 숫자 부분에 문자열 처리를 해줍니다.

2. 연산자 버튼 생성

42	-	operatorList = [
43	-	'*', '/',
44	-	'+', '-',
45	-	'(', ')',
46	-	'C',
47	-]
48	-	# Operator Buttons
49	-	for button in operatorList[:]:
50	-	self.mulButton = Button(button, self.buttonClicked)
42	+	button = {'mulButton': '*', 'divButton': '/', 'addButton': '+', 'subButton': '-',
43	+	'lparButton': '(', 'rparButton': ')', 'clearButton': 'C'}
44	+	
45	+	for key, value in button.items():
46	+	self.key = Button(str(value), self.buttonClicked)

- 변수명을 키, 연산자를 값으로 가지는 button 딕셔너리를 만들고 for 문을 사용하여 연산자 버튼을 생성하면 정상적으로 작동 할 줄 알았는데 오류가 뜹니다. 잘못된 부분을 더 고민해서 수정해야 할 것 같습니다.

3. 예외 처리

```
114      87      def buttonClicked(self):
115      88          button = self.sender()
116      89          key = button.text()
117      90          if key == '=':
118      -          result = str(eval(self.display.text()))
119      91      +          try:
120      92      +              result = str(eval(self.display.text()))
121      93      +          except:
122      94      +              result = 'Error!'
123      95          self.display.setText(result)
```

-eval 함수로 연산을 할 때 올바른 수식이 사용되지 않았을 때 프로그램이 다운되지 않도록 예외처리를 해주어 0 으로 나누었을때의 ZeroDivisionError, 현존 리스트 이상의 index 를 가리켜 나타나는 IndexError 등을 예방할 수 있습니다.