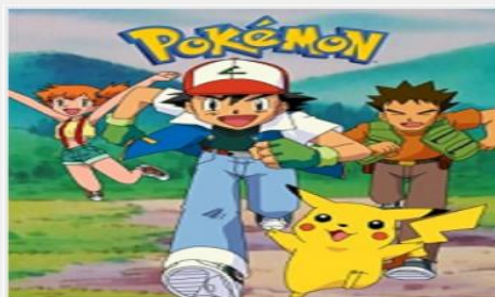


# 파이썬 tkinter 적용해서 만든 온라인 만화책 대여 Kiosk 서비스 제작

---

훈련생 : 탁우근

# UI 소개



포켓몬스터 : 500 원



**디지몬 어드벤처 : 500 원**



가정교사 히트맨 리본 : 500원



귀멸의 칼날 : 1000 원



개구리 종사 케로로: 1000원



**짱구는 못말려 : 1000 원**

## 전체 취소

중

대여

회원 이름 : Developking

회원 전화 : 010-1111-1111

금액 10000원

수량11권

1. 이미지 버튼
2. 전체 취소
3. 대여 버튼
4. 회원 이름
5. 회원 전화번호
6. 총 금액
7. 책 수량
8. 보기 목록
9. 종료 버튼

# 버튼 이미지 삽입

---

```
#이미지 삽입
```

```
menu1 =tkinter.PhotoImage(file="item1.png")
```

```
menu2 =tkinter.PhotoImage(file="item2.png")
```

```
menu3 =tkinter.PhotoImage(file="item3.png")
```

```
menu4 =tkinter.PhotoImage(file="item4.png")
```

```
menu5 =tkinter.PhotoImage(file="item5.png")
```

```
menu6 =tkinter.PhotoImage(file="item6.png")
```

# 이벤트 연결

---

#이벤트 연결

```
btn_1 = tk.Button(frame1, image=menu1, command=lambda :add('포켓몬스터'), width=270, height=250)
btn_2 = tk.Button(frame1, image=menu2, command=lambda :add('디지털어드벤처'), width=270, height=250)
btn_3 = tk.Button(frame1, image=menu3, command=lambda :add('가정교사 히트맨 리본'), width=270, height=250)
btn_4 = tk.Button(frame1, image=menu4, command=lambda :add('귀멸의 칼날'), width=270, height=250)
btn_5 = tk.Button(frame1, image=menu5, command=lambda :add('개구리중사 케로로'), width=270, height=250)
btn_6 = tk.Button(frame1, image=menu6, command=lambda :add('짱구는 못말려'), width=270, height=250)
btn_7 = tk.Button(frame1, text="대여 ", command=btn_lent, width=10, height=2)
btn_8 = tk.Button(frame1, text="전체 취소", command=clear, width=10, height=2)
btn_9 = tk.Button(frame1, text="종료", command=exit, width=10, height=2)
```

# 이벤트 위치

---

```
#위치
#row = 행 , # column = 줄 #padx = 가로 여백, pady = 세로 여백
btn_1.grid(row=0, column=0, padx=5, pady=5)
btn_2.grid(row=0, column=1, padx=5, pady=5)
btn_3.grid(row=0, column=2, padx=5, pady=5)
btn_4.grid(row=1, column=0, padx=5, pady=5)
btn_5.grid(row=1, column=1, padx=5, pady=5)
btn_6.grid(row=1, column=2, padx=5, pady=5)
btn_7.grid(row=2, column=2, padx=5, pady=5)
btn_8.grid(row=2, column=0, padx=5, pady=5)
btn_9.grid(row=2, column=1, padx=5, pady=5)
```

# 회원의 유효성 검사

---

```
# 회원의 유효성 검사

if name == "":
    tk.messagebox.showerror("확인", "이름을 입력해주세요")
    entry1.forcus()
    return

if hp == "":
    tk.messagebox.showerror("확인", "휴대폰 번호를 입력해주세요")
    entry2.forcus()
    return
```

# 메뉴가 추가되는 함수\_add(item)

```
#주문 버튼을 눌렀을때 해당 메뉴가 추가되는 함수
def add(item):
    #전역변수로 써야하기 때문에 global sum
    global sum, count, name
    if item not in price:
        print("no comic_book")
    this_price = price.get(item)

    #가격과 수량 가져오기
    sum += this_price
    count += 1

    #주문 메뉴 추가하기
    order.append(item)

    #order2의 수량을 바꿔주는 list
    order2[item] += 1

    #해당 내용 넣어주기
    textarea.insert(tk.INSERT, item+"\n")
    label1["text"] = "금액" + str(sum) + "원"
    label4["text"] = "수량" + str(count) + "권"
```

# 조건이 충족 시 데이터 전송

---

#message박스로 대여 여부 확인

```
msgbox = tk.messagebox.askquestion("확인", "정말로 대여 하시겠습니까?\n대여기간 : 3일 입니다.")
```

#msgbox가 YES이면 주문전송

```
if msgbox == 'yes':
```

```
    check = name + "님" + ":" + "총" + count + "권" + "대여" + "하셨습니다."
```

```
    tk.messagebox.showinfo("확인", check)
```

```
    tk.messagebox.showinfo("확인", "감사합니다. 또 이용해주세요.")
```

#DB에 입력된 데이터들 저장

```
add_rentlist(user_name, user_tel, total, count, list)
```

```
clear()
```



# 모두 초기화 : clear()

```
# 모두 초기화
def clear():
    #global = 전역변수를 사용 하는 변수 정의
    global sum, order, order2, textarea, entry1, entry2, count
    textarea.delete('1.0', tk.END)
    label1['text'] = "금액 0원"
    label4['text'] = "수량 0권"

    sum = 0
    order = []
    count = 0
    order2 = {'포켓몬스터':0, '디지털어드벤처':0, '가정교사 히트맨 리본':0, '귀멸의 칼날':0, '개구리중사 케로로':0, '짱구는 못말려':0}
    entry1.delete('0', tk.END)
    entry2.delete('0', tk.END)
    entry1.focus()
```

# DB 테이블, 시퀀스 생성

```
--제품 구매 테이블
SELECT * FROM RENT;
DROP TABLE RENT;
CREATE TABLE RENT(
    user_number number primary key, --회원 번호
    user_name varchar2(30),         --회원 이름
    user_tel varchar2(20),          --회원 전화번호
    total number,                   --total 금액
    count number,                   --총 수량
    list varchar2(3000)             --빌린 목록 보기
);

SELECT * FROM RENT;

DROP SEQUENCE SEQ_RENT_USERNO;
create sequence SEQ_RENT_USERNO
minvalue 1 --시퀀스 최소 숫자
maxvalue 9999 --시퀀스 최대 숫자
increment by 1 --증가치
start with 1; --시작숫자
```

# DB 저장 : add\_rentlist(\*)

---

```
#DB 테이블에 저장
def add_rentlist(user_name, user_tel, total, count, list):
    conn = oci.connect('SCOTT/TIGER@localhost:1521/XE')
    print(conn.version)
    # 2) 커서(Cursor) / PreparedStatement와 비슷
    cursor = conn.cursor()

    # 3) SQL 문장
    sql = '''
        INSERT INTO RENT (user_number, user_name, user_tel, total, count, list)
        VALUES (SEQ_RENT_USERNO.nextval, '{}', '{}', {}, {}, '{}')
    '''.format(user_name, user_tel, total, count, list)

    # 4) SQL 실행
    cursor.execute(sql)

    # 5) 커서 닫기
    cursor.close()

    # 6) 커밋
    conn.commit()

    # 7) 연결 닫기
    conn.close()
```

## 결과 : 확인

Output		SCOTT.RENT			
7 rows		Tx: Manual		DDL	
USER_NUMBER		USER_NAME		USER_TEL	
9 이승기		010-1313-2121		5000	
10 버닝타		010-1111-9988		5000	
11 옥은님		010-9282-8811		5000	
6 Developking		010-1111-1111		10000	
7 탁우근		010-9999-9999		18000	
8 SIAT		011-2932-2323		13000	
12 이민기		010-1212-1234		15000	

---

감사합니다.

