

## What's New

- بدلاً من كل المنطق في الحلقة `game(score, computer_score)` استخدام الدوال: تعريف دالة.
- (المعاملات والإرجاع المتعدد: تمرير المتغيرات للدالة وإرجاع قيمتين (نتيجتك ونتيجة الكمبيوتر).
- تتبع النتائج: عرض لوحة النتائج في كل جولة.
- عند دمج النصوص مع الأرقام `str()` تحويل الأنواع للطباعة: استخدام.

## Topic 1: Functions — Parameters and Return Values

### تعريف Game function

#### 1. تعريف الدالة وتلقي النتائج:

```
def game(score, computer_score):
    # تطلب من المستخدم إدخال اختياره
    user_input = input("enter your choice :")
    # الكمبيوتر يختار بشكل عشوائي من القائمة
    computer_choice = random.choice(choices)

    # مقارنة الاختيارات:
    if user_input == computer_choice:
        print("🤝 Draw") # تعادل

    # حالات فوز المستخدم
    elif (
        user_input == "📄" and computer_choice == "🔪" or
        user_input == "🔪" and computer_choice == "✂️" or
        user_input == "✂️" and computer_choice == "📄"
    ):
        score = score + 1
        print("you WOONN 🌟🌟")
        print("Computer Lose 😞")

    # حالات فوز الكمبيوتر
    elif (
        user_input == "✂️" and computer_choice == "🔪" or
        user_input == "📄" and computer_choice == "✂️" or
        user_input == "🔪" and computer_choice == "📄"
    ):
        computer_score = computer_score + 1
        print("you LOSE 😞")
        print("computer WOONN 🌟🌟")

    # معالجة أي إدخال غير صحيح
    else:
        print("choose only from : " + str(choices))

    # تُعيد النتائج الجديدة (نتيجة اللاعب ونتيجة الكمبيوتر)
    return score, computer_score
```

### شرح الأجزاء:

- `input()`: أو "📄" تطلب من المستخدم إدخال رمزه (مثلاً "🔑" أو "🔒")
- `random.choice(choices)`: الكمبيوتر يختار رمزاً بشكل عشوائي من القائمة
- (الشروط تحدد النتيجة (فوز/خسارة/تعادل
- إذا كان الإدخال غير صحيح، تظهر رسالة للمستخدم
- الدالة تعيد نتيجتك ونتيجة الكمبيوتر بعد كل جولة

### أفكار أساسية

- يتم تمريرها للدالة `score, computer_score`: **المعاملات**
- `tuple`: **الإرجاع المتعدد**: الدالة تعيد قيمتين باستخدام
- **الاستخدام**:

```
score, computer_score = game(score, computer_score)
```

## Topic 2: Scoreboard and String Conversion

لوحة النتائج وتحويل الأنواع للطباعة

```
print(" 😊 Your Score is : " + str(score) + " | " + " 🖥 Computer Score is : " + str(computer_score))
```

- ؟: عند دمج أرقام مع نصوص يجب تحويل الأرقام إلى نص `str()` لماذا
- (بديل أفضل اختياري):

```
print(f" 😊 Your Score is : {score} | 🖥 Computer Score is : {computer_score}")
```

## Topic 3: Main Loop — Controlling Rounds

while التحكم بعدد الجولات باستخدام

```
number_of_games = 5
score = 0
computer_score = 0

while number_of_games > 0: # لتجنب جولة إضافية
    score, computer_score = game(score, computer_score)
    print(" 😊 Your Score is : " + str(score) + " | " + " 🖥 Computer Score is : " + str(computer_score))
    number_of_games = number_of_games - 1
```

- نصيحة: استبدل  $0 < 0$  بـ  $0 = 0$  لتجنب لعب جولة سادسة عندما تبدأ من 5

## Topic 4: Input Validation and UX

التحقق من صحة الإدخال وتجربة المستخدم

- استخدم قائمة بالاختيارات المسموحة.
- اطبع الاختيارات للمستخدم بشكل واضح.

```
choices = ["✂️", "🪨", "📄"]
user_input = input(f"Choose one {'', '.join(choices)}: ")
if user_input not in choices:
    print("Please choose a valid option!")
```

## Full Polished Example

مثال كامل ومنسق للعبة

```
import random

choices = ["✂️", "🪨", "📄"]

def game(score, computer_score):
    user_input = input(f"enter your choice ({', '.join(choices)}): ")
    computer_choice = random.choice(choices)

    print("The computer choice is : " + computer_choice)

    if user_input == computer_choice:
        print("🤖 Draw")

    elif (
        user_input == "📄" and computer_choice == "🪨" or
        user_input == "🪨" and computer_choice == "✂️" or
        user_input == "✂️" and computer_choice == "📄"
    ):
        score += 1
        print("you WOONN 🌟🌟 ")
        print("Computer Lose 😞 ")

    elif (
        user_input == "✂️" and computer_choice == "🪨" or
        user_input == "📄" and computer_choice == "✂️" or
        user_input == "🪨" and computer_choice == "📄"
    ):
        computer_score += 1
        print("you LOSE 😞")
        print("computer WOONN 🌟🌟")
```

```
        else:
            print("choose only from : " + ", ".join(choices))

        return score, computer_score

number_of_games = 5
score = 0
computer_score = 0

while number_of_games > 0:
    score, computer_score = game(score, computer_score)
    print(f" 😊 Your Score is : {score} | 📊 Computer Score is : {computer_score}")
    number_of_games -= 1
```

## Key Concepts Covered

المفاهيم الرئيسية التي تم تغطيتها:

- 1. **Functions:** تعريف الدوال، المعاملات، والقيم المعادة.
- 2. **Multiple Return:** إرجاع أكثر من قيمة من الدالة.
- 3. **Scorekeeping:** تحديث النتائج وعرض لوحة النتائج.
- 4. **Loops:** التحكم بعدد الجولات باستخدام `while`.
- 5. **String Conversion:** عند الطباعة f-strings أو `str()` استخدام.
- 6. **Input Validation:** التحقق من صحة إدخال المستخدم.

## Advanced: Common Pitfalls and Best Practices

أخطاء شائعة ونصائح مهمة

| Concept             | Common Mistake                             | Best Practice                                | Solution   |
|---------------------|--|--|--|
| Shadowing built-ins | <code>list</code> تسمية متغير باسم         | استخدم اسمًا واضحًا مثل <code>choices</code> | <code>choices = ["🔪", "🎲", "📊"]</code>             |
| Loop condition      | <code>while number_of_games &gt;= 0</code> | استخدم <code>0 &lt;</code>                   | <code>while number_of_games &gt; 0:</code>         |
| Emoji matching      | رمز غير صحيح في المقارنة                   | نسخ نفس الرموز بالضبط                        | <code>"🔪", "🎲", "📊"</code> تأكد من                 |
| String + int        | دمج أرقام مع نصوص مباشرة                   | f-string حوّل أو استخدم                      | <code>str(score)</code> أو <code>f"{score}"</code> |
| Return handling     | نسيان استقبال قيم الإرجاع                  | استقبل كل القيم                              | <code>score, computer_score = game(...)</code>     |

تجنب هذه الأخطاء لبرمجة أنظف وتجربة لعب أفضل