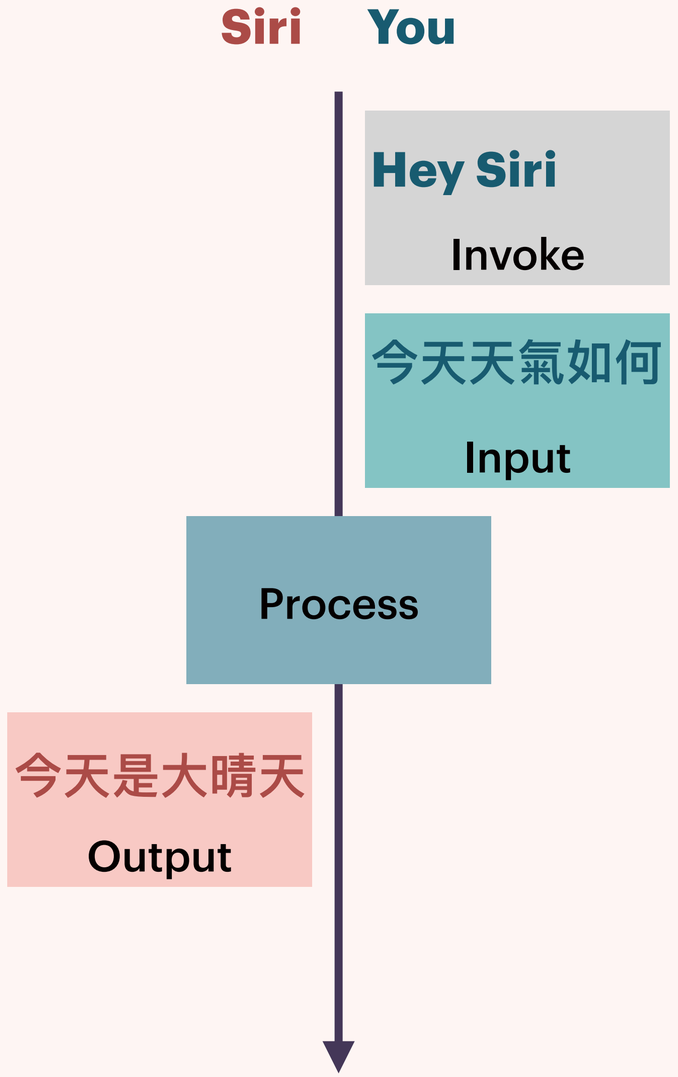
函式 FUNCTION

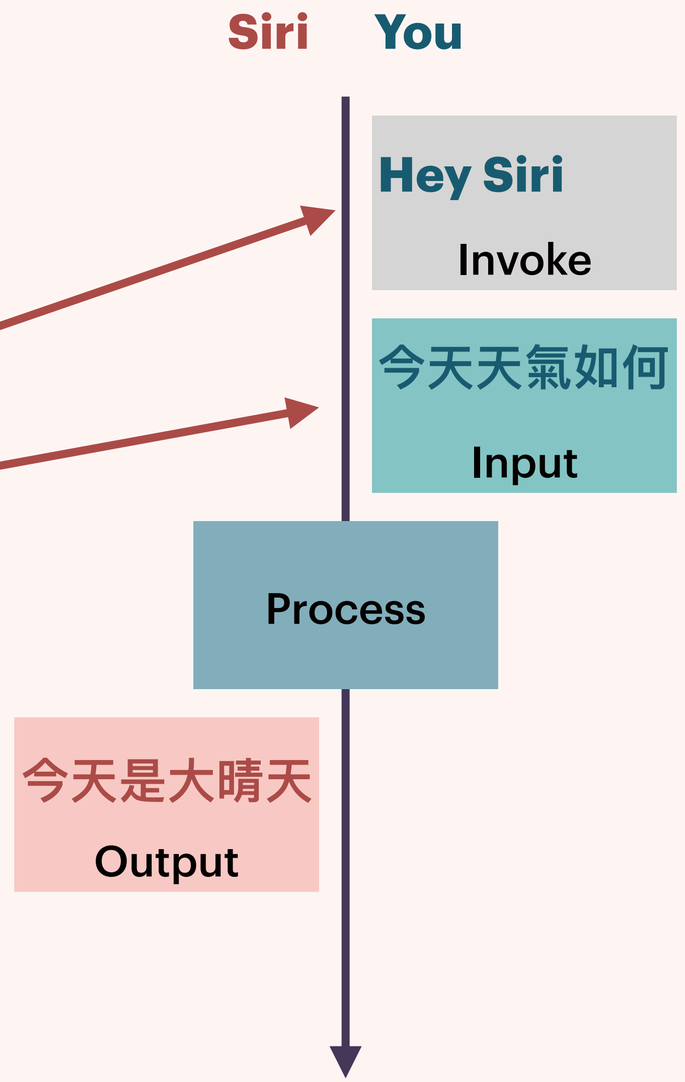
Siri	You
	Hey Siri 今天天氣如何
今天是大晴天	
	Hey Siri 說個笑話
不要再講啦	
	Hey Siri 我想想看要幹嘛
我不懂你的意思	


9:41
有什麼我可以
幫忙的嗎？


Siri
You
Hey Siri
Invoke
今天天氣如何
Input
Process
今天是大晴天
Output

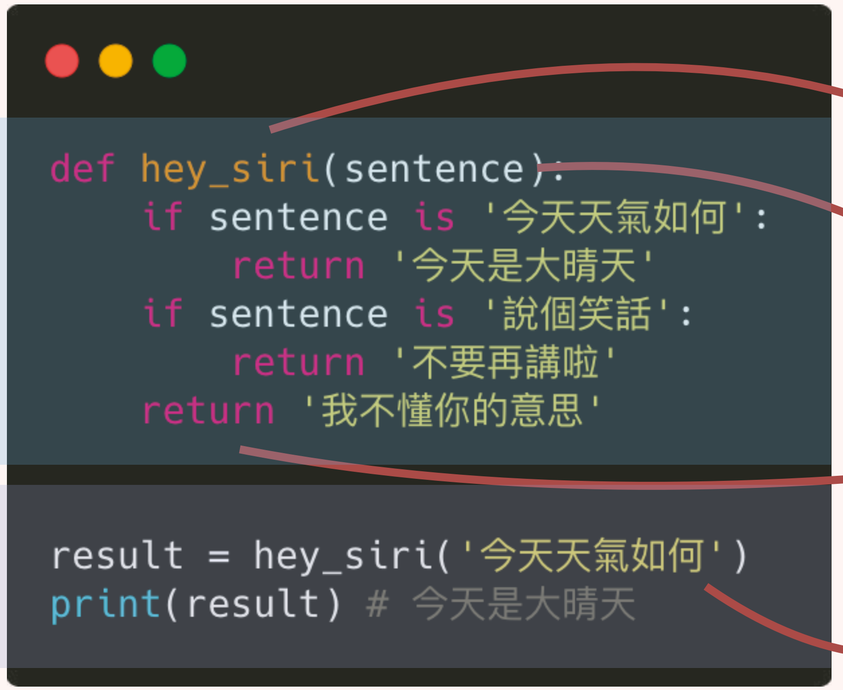
```python
def hey_siri(sentence):
    if sentence is '今天天氣如何':
        return '今天是大晴天'
    if sentence is '說個笑話':
        return '不要再講啦'
    return '我不懂你的意思'


result = hey_siri('今天天氣如何')
print(result) # 今天是大晴天
```



定義
Define

調用
Invoke



```
def hey_siri(sentence):
    if sentence is '今天天氣如何':
        return '今天是大晴天'
    if sentence is '說個笑話':
        return '不要再講啦'
    return '我不懂你的意思'

result = hey_siri('今天天氣如何')
print(result) # 今天是大晴天
```

- 名稱 **Name**
- (內部)參數 **Parameters**
- 回傳值 **Return value**

(外部)參數 **Arguments**

函式好處

- **Reusable - 可再利用**
- **Maintainable - 可維護**
- **Implementable - 可實作**

POSITIONAL V.S. KEYWORD ARGUMENTS

```python
def add(x1, x2, x3, x4, x5):
    return x1 + x2 + x3 + x4 + x5
```

```python
# positional args
add(1, 2, 3, 4, 5)
```

```python
# keyword args
add(x1=1, x2=2, x3=3, x4=4, x5=5)
```

```python
# mixed
add(1, 2, x3=3, x4=4, x5=5)
```

```python
def echo(x1, x2, *args, **kwargs):
    print(x1, x2, args, kwargs)
```

```python
echo(1, 2, 3, 4, x=5, y=6)
# 1, 2, [3, 4], {"x": 5, "y": 6}

echo(1, 2)
# 1, 2, [], {}

echo(x1=1, x2=2, x3=3)
# 1, 2, [], {"x3": 3}
```

1. **先 Positional 再 Keyword**
2. **Positional args 用 * 打包 tuple**
3. **Keyword args 用 ** 打包成 dict**