Loops

5. حلقه ها

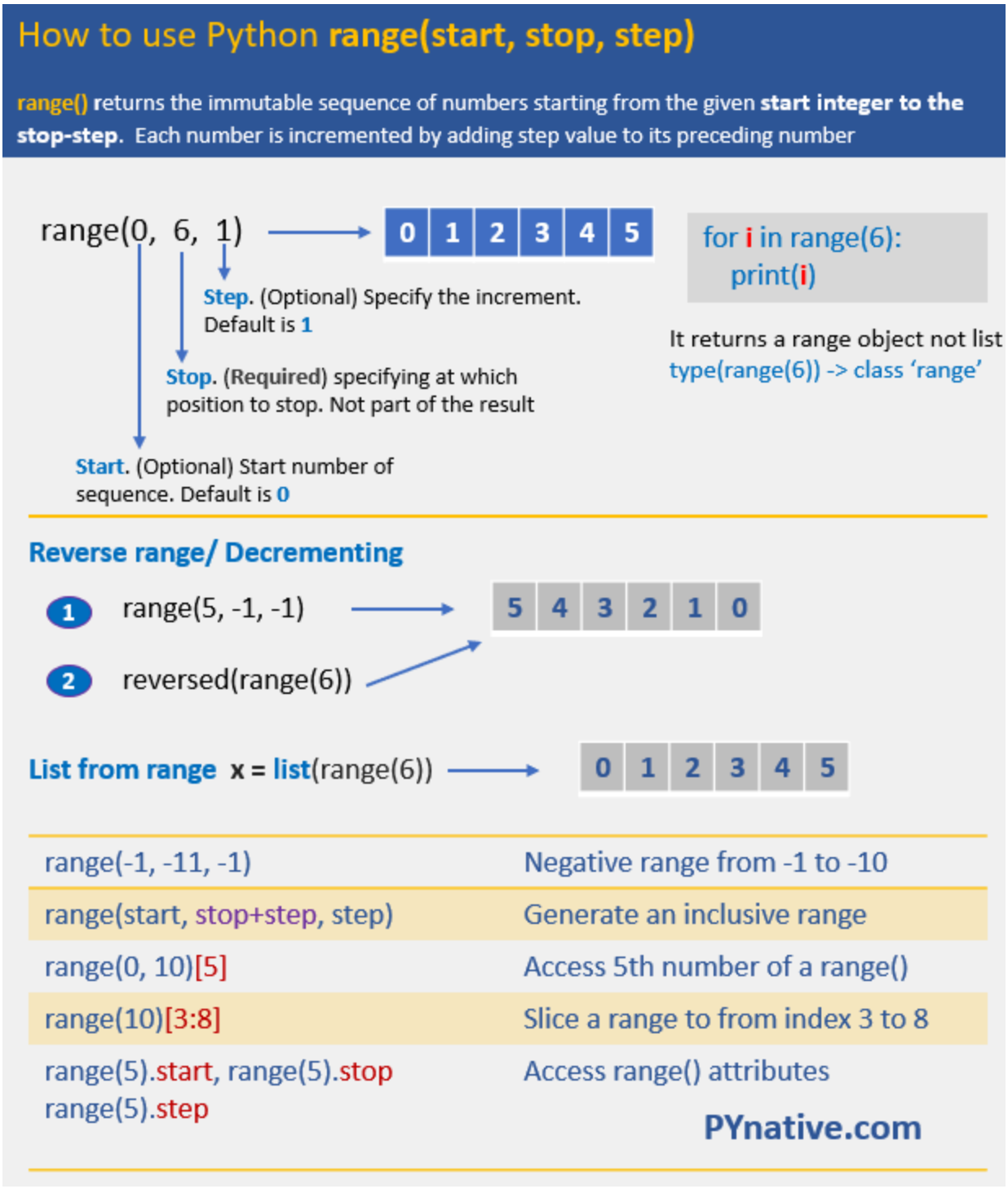
For Loop (حلقه For)

از حلقه `for` برای پیمایش در یک بازه یا مجموعه استفاده کنید.

برای اجرای مکرر مجموعه‌ای از کدها به تعداد مشخص، می‌توان از تابع `()range` استفاده کرد. این تابع یک دنباله از اعداد ایجاد می‌کند که به‌صورت پیش‌فرض:

- از عدد `0` شروع می‌شود،
- با گام `1` افزایش می‌یابد،
- و پیش از رسیدن به عدد انتهایی مشخص‌شده متوقف می‌شود.

سینتکس این تابع به صورت زیر است : range(start, stop, step_size)




```
# Looping from 0 to 4
for i in range(5):
    print("Number:", i)  # Output: 0, 1, 2, 3, 4
```

زمانی که از تابع `range()` استفاده می‌کنیم، چه اتفاقی می‌افتد؟

تابع `range()` در واقع یک **دنباله (Sequence)** از اعداد تولید می‌کند. این دنباله به‌صورت دینامیک (در زمان اجرا) ایجاد می‌شود و به‌جای ذخیره‌شدن به‌عنوان یک لیست واقعی در حافظه، تنها بازه و گام‌های تولید اعداد را نگهداری می‌کند.

در صورتی که این دنباله را به لیست تبدیل کنیم، اعداد به‌صورت یک لیست واقعی در حافظه ذخیره می‌شوند. برای مثال:

```
numbers = list(range(5))
print(numbers)
# output: [0, 1, 2, 3, 4]
```

در این صورت، خروجی `[0, 1, 2, 3, 4]` خواهد بود که یک لیست واقعی است.

بدون استفاده از `list()` ، تابع `range()` تنها یک دنباله از اعداد را تولید می‌کند که به‌صورت مستقیم می‌توان از آن در حلقه‌ها یا دیگر عملیات‌ها استفاده کرد. مثلاً:

```
for i in range(5):
    print(i)
```

تحلیل عملکرد:

۱. با اجرای `range(5)` ، دنباله‌ای از اعداد از `0` تا `4` (یعنی `n-1`) به‌طور دینامیک تولید می‌شود.

۲. در هر گام از حلقه، مقدار `i` یکی از مقادیر این دنباله را دریافت می‌کند:

- ابتدا `i = 0` ، سپس `i = 1` ، و به همین ترتیب تا `i = 4` .

۳. زمانی که همه مقادیر پیمایش شدند، حلقه به پایان می‌رسد.

تابع len()

تابع `len()` تعداد عناصر موجود در یک شیء را بازمی‌گرداند. این تابع برای انواع مختلفی از اشیاء در پایتون قابل استفاده است، مانند:

- **رشته‌ها (string)**: تعداد کاراکترهای موجود در رشته را برمی‌گرداند.
- **لیست‌ها (list)**: تعداد عناصر موجود در لیست را بازمی‌گرداند.
- **سایر انواع مجموعه‌ها (collections)**: تعداد اعضای آن‌ها را ارائه می‌دهد.

مثال: محاسبه تعداد کاراکترهای یک رشته

اگر شیء ورودی یک رشته باشد، تابع `len()` تعداد کاراکترهای موجود در آن را بازمی‌گرداند. به مثال زیر توجه کنید:

```
myString = "Hello"
print(len(myString))  # output: 5
```

حلقه While

از حلقه `while` برای اجرای کد تا زمانی که یک شرط برقرار باشد استفاده کنید.

```
# Countdown loop
count = 3
while count > 0:
    print("Counting down:", count)
    count -= 1  # Decrease the count by 1
```