

<[파이썬 알고리즘 탐험] 제6장. 숫자 변신의 귀재! 진법 변환 마법!>

선생님: (여러 나라의 화폐를 보여주며) 애들아, 안녕! 우리가 '백 원'이라고 부르는 돈은 숫자 100으로 표현하지? 이걸 열 개씩 묶어서 세는 10진법 세상의 이야기야. 그런데 만약 두 개씩 묶어서 세는 세상이 있다면, 이 100이라는 숫자는 어떻게 표현될까? 오늘은 이렇게 숫자를 다른 '진법'으로 표현하는 방법을 배울 거야.

<1. 숫자 분해 마법! - 10진수를 자릿수별로 나누기>

선생님: 먼저, 우리가 익숙한 10진수 숫자를 각 자릿수의 숫자로 분해해서 리스트에 담아보는 함수를 만들어볼까? 예를 들어 숫자 532를 [5, 3, 2] 이렇게 말아야.

코딩새싹 🌱: 으음... 가장 마지막 자리 숫자부터 하나씩 꺼내면 될 것 같아요! 532에서 2를 꺼내고, 그 다음엔 3을 꺼내고...

선생님: 좋은 생각이야! 어떤 숫자의 가장 마지막 자리(일의 자리) 숫자를 알아내려면 숫자 % 10 (10으로 나눈 나머지) 마법을 쓰면 되지? 그리고 그 마지막 자리를 없애려면 숫자 // 10 (10으로 나눈 몫) 마법을 쓰면 되고!

532 => [5, 3, 2]

```
def deci_num(a):
```

```
    result = [] # 각 자릿수 숫자를 담을 빈 리스트
```

```
    while a > 0: # 숫자가 0보다 클 동안 계속 반복!
```

```
        remainder = a % 10 # 1. 마지막 자릿수를 구한다 (예: 532 % 10 = 2)
```

```
        result.append(remainder) # 2. 결과 리스트에 추가! (지금은 [2], 다음엔 [2, 3], 그 다음엔 [2, 3, 5] 순서로)
```

```
        a = a // 10 # 3. 마지막 자릿수를 없앤다 (예: 532 // 10 = 53)
```

```
    # 숫자를 오른쪽부터 꺼냈으니, 원래 순서대로 보려면 리스트를 뒤집어야 해!
```

```
    result.reverse()
```

```
    return result
```

```
print("389021을 분해하면:", deci_num(389021))
```

궁금이 💡: while a > 0: 조건 때문에 a가 0이 되면 반복이 멈추는군요! 그리고 result.reverse()로 마지막에 순서를 뒤집어주는 게 핵심이네요!

<2. 컴퓨터의 언어 배우기! - 10진수를 2진수로!>

선생님: 컴퓨터는 사실 0과 1밖에 모르는 아주 단순한 친구란다. 그래서 모든 숫자를 0과 1로만 표현하는 2진법을 사용하지. 우리가 만든 `dec_i_num` 함수를 살짝만 바꾸면 10진수를 2진수로 바꾸는 마법도 부릴 수 있어!

코딩새싹 🌱: 어? 정말요? 어떻게 바꾸죠?

선생님: 10진법에서는 10으로 나누고 나머지를 구했지만, 2진법에서는 2로 나누고 나머지를 구하면 되는 거야! 나머지는 항상 0 아니면 1이 나오겠지?

십진수를 이진수로 바꾸는 함수 정의, 호출

```
def deci_to_bi(num):
```

```
    result = []
```

```
    while num > 0:
```

```
        remainder = num % 2 # 2로 나눈 나머지를 구한다!
```

```
        result.append(remainder)
```

```
        num = num // 2 # 2로 나눈 몫을 다음 숫자로!
```

```
    result.reverse() # 똑같이 뒤집어주기!
```

```
    return result
```

```
print("\n숫자 8을 2진수로 바꾸면:", deci_to_bi(8)) # [1, 0, 0, 0] 이 나오겠지? ( $8 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$ )
```

궁금이 💡: 와! 10 대신 2를 썼을 뿐인데 2진수로 바뀌었어요! 그럼 혹시... 5진수나 16진수 같은 다른 진법으로도 바꿀 수 있나요?

<3. 만능 진법 변환기! - 10진수를 N진수로!>

선생님: 물론이지! 바로 그게 `dec_i_to_ndigit(숫자, n)` 함수야. `n` 자리에 우리가 원하는 진법 숫자를 넣어주면 된단다!

십진수 a를 n진수로 바꾸는 함수

```
def deci_to_ndigit(a, n_base): # n_base가 우리가 변환하고 싶은 진법!
```

```
    result = []
```

```

while a > 0:
    remainder = a % n_base # n_base로 나눈 나머지를 구한다!
    result.append(remainder)
    a = a // n_base # n_base로 나눈 몫을 다음 숫자로!

result.reverse()

return result

```

```

print("\n숫자 32를 5진수로 바꾸면:", deci_to_ndigit(32, 5)) # [1, 1, 2] (32 = 1*5^2 + 1*5^1 + 2*5^0 = 25

```

<4. 다시 우리 세상으로! - N진수를 10진수로!>

선생님: 자, 그럼 반대로 다른 진법으로 표현된 숫자를 우리가 쓰는 10진수로 다시 되돌리는 마법도 있어야겠지?

예를 들어 2진수 1101은 10진수로 얼마일까?

코딩새싹 🌱: 으음... 각 자리가 나타내는 실제 값을 더하면 될 것 같아요! 2진수 1101은...

오른쪽부터 $1 * (2의 0제곱) + 0 * (2의 1제곱) + 1 * (2의 2제곱) + 1 * (2의 3제곱)$ 아닌가요?

즉, $1*1 + 0*2 + 1*4 + 1*8 = 1 + 0 + 4 + 8 = 13$ 이요!

선생님: (엄지를 척 들며) 정확해! 각 자릿수에 해당하는 숫자와, 그 자리가 나타내는 (진법의 거듭제곱) 값을 곱해서 모두 더해주면 된단다! 이 원리를 코드로 옮겨보자.

n진수 a를 십진수로 바꾸는 함수를 만들어 보세요.

(두 번째 버전 코드가 더 직관적이어서 그걸로 설명할게!)

```

def to_deci(num_str, n_base):
    # num_str은 '1101'처럼 다른 진법의 숫자를 "문자열" 형태로 받기로 약속하자!
    digits = list(str(num_str)) # 각 자릿수를 분리해서 리스트로 만들. ['1', '1', '0', '1']
    digits.reverse() # 오른쪽 자릿수부터 처리하기 위해 뒤집는다! ['1', '0', '1', '1']

    decimal_value = 0

    # enumerate를 쓰면 번호표(power_idx)와 내용물(digit_char)을 함께 받을 수 있지!
    for power_idx, digit_char in enumerate(digits):
        digit_value = int(digit_char) # 문자 '1'을 숫자 1로 변환

```

```

# power_idx가 0일때 n_base^0, 1일때 n_base^1... 이렇게 자릿값을 곱해준다!
decimal_value += (digit_value * (n_base ** power_idx))

return decimal_value

print("\n2진수 '1101'을 10진수로 바꾸면:", to_deci("1101", 2))
print("5진수 '112'를 10진수로 바꾸면:", to_deci("112", 5))

```

궁금이 💡: `digits.reverse()`로 리스트를 뒤집으니까, `enumerate`로 나오는 `power_idx`가 자연스럽게 0제공, 1제공, 2제공... 순서로 딱 맞게 떨어지네요! 정말 똑똑한 방법 같아요!

선생님: (미소 지으며) 그렇지? 오늘 우리는 10진수를 여러 진법으로 바꾸는 마법, 그리고 다른 진법의 수를 다시 10진수로 되돌리는 마법까지 배웠어. 특히 '나머지'와 '몫'을 반복적으로 사용하는 아이디어, 그리고 각 자릿수의 '가중치(거듭제곱)'를 활용하는 아이디어는 진법 변환의 핵심이란다! 이제 여러분은 숫자 변신의 귀재가 되었으니, 컴퓨터와 더 깊은 대화도 나눌 수 있을 거야! 오늘 정말 어려운 내용을 탐구하느라 수고 많았어! 100