

02 데이터 분석 기초 - 넘파이

AI 에이전트 개발

데이터 분석 기초

원티드랩

- [소개](#)
- [1. 배열](#)
 - [1\) 생성](#)
 - [2\) 속성](#)
 - [3\) 변환](#)
 - [4\) 통계](#)
- [2. 사용 예제](#)
 - [1\) 데이터 생성](#)
 - [2\) 이미지 처리](#)

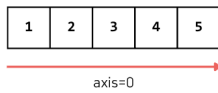
소개



- Numerical Python의 줄임말
- 고성능 과학 계산과 고차원 배열에 필요한 수학 패키지
- 빠른 수치 계산을 위해 C 언어로 만들어진 Python 라이브러리
- 벡터와 행렬 연산에 매우 편리한 기능 제공

1. 배열

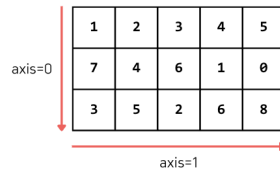
1차원



ndim: 1
shape: (5,)

```
[1, 2, 3, 4, 5]
```

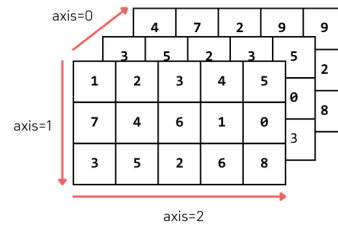
2차원



ndim: 2
shape: (3, 5)

```
[
  [1, 2, 3, 4, 5],
  [7, 4, 6, 1, 0],
  [3, 5, 2, 6, 8]
]
```

3차원



ndim: 3
shape: (3, 3, 5)

```
[
  [
    [1, 2, 3, 4, 5],
    [7, 4, 6, 1, 0],
    [3, 5, 2, 6, 8]
  ],
  [
    [3, 5, 2, 3, 5],
    [7, 4, 6, 1, 0],
    [3, 5, 2, 6, 3]
  ],
  [
    [4, 7, 2, 9, 9],
    [7, 4, 6, 1, 2],
    [3, 5, 2, 6, 8]
  ]
]
```

1) 생성

함수	설명
<code>np.array(데이터)</code>	데이터로 배열 생성
<code>np.zeros(크기)</code>	모든 요소가 0인 배열 생성
<code>np.ones(크기)</code>	모든 요소가 1인 배열 생성
<code>np.arange(start, end, x)</code>	start부터 end-1까지 x 간격으로 배열 생성

함수	설명
<code>np.linspace(x, y, n)</code>	x와 y 사이를 n등분한 배열 생성

2) 속성

속성	설명
<code>배열.shape</code>	배열의 크기
<code>배열.dtype</code>	배열의 데이터 타입
<code>배열.ndim</code>	배열의 차원
<code>배열.size</code>	배열 전체 요소 개수

3) 변환

함수	설명
<code>배열.reshape(차원)</code>	배열의 차원 변경
<code>배열.flatten()</code>	다차원 배열을 1차원으로 변경
<code>배열.transpose()</code>	배열의 행

4) 통계

`arr = np.array([1, 2, 3])` 이라고 할 때 `np.통계함수(arr)`, `arr.통계함수()` 모두 사용 가능

함수	설명
<code>np.mean(array)</code>	배열의 평균
<code>np.sum(array)</code>	배열의 합
<code>np.max(array)</code>	배열의 최대값
<code>np.min(array)</code>	배열의 최소값
<code>np.var(array)</code>	배열의 분산
<code>np.std(array)</code>	배열의 표준편차

차원이 큰 경우에 `axis` 를 설정하여 요약할 수도 있다.

```
arr = np.array([
    [
        [1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]
    ],
    [
        [10, 11, 12],
        [13, 14, 15],
        [16, 17, 18]
    ]
])

ndim: 3
shape: (2, 3, 3)
```

```
np.sum(arr, axis=0)

[
    [11, 13, 15],
    [17, 19, 21],
    [23, 25, 27]
]

ndim: 2
shape: (3, 3)
```

```
np.sum(arr, axis=1)

[
    [12, 15, 18],
    [39, 42, 45]
]

ndim: 2
shape: (2, 3)
```

```
np.sum(arr, axis=2)

[
    [6, 15, 24],
    [33, 42, 51]
]

ndim: 2
shape: (2, 3)
```

2. 사용 예제

1) 데이터 생성

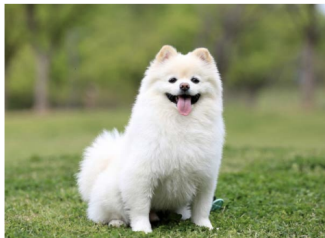
난수 생성

함수	설명
<code>np.random.seed(숫자)</code>	랜덤 시드 고정
<code>np.random.rand(num)</code>	0~1 사이의 랜덤 실수 값 num개 생성
<code>np.random.randint(start, end, num)</code>	start~end-1 사이의 랜덤 정수 값 num개 생성
<code>np.random.randn(num)</code>	정규분포 난수 값 num개 생성

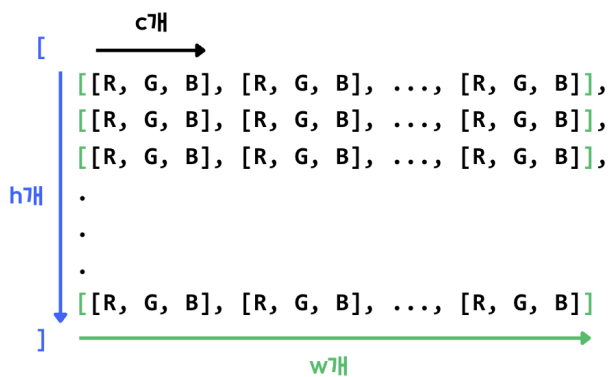
2) 이미지 처리

- 이미지: 픽셀로 구성된 숫자 배열이며 0~255 사이의 값으로 표현된다.
- 배열 연산을 통해 이미지를 변환할 수 있다.
- JPG의 경우 RGB로 이루어져 있지만, PNG의 경우 RGBA로 이루어져 있다.

이미지 이해하기(ex. jpg)



ndim: 3
shape: (h, w, c)



이미지 불러오기

- 이미지 관리 도구: `pillow`

```
uv add pillow
```

```
from PIL import Image
import numpy as np

image = Image.open("파일경로")
image_arr = np.array(image)

print(image_arr.shape)
```