# 07 파이썬 활용(내장함수)

- 내장함수

  ✓ 파이썬에서 제공하는 이미 만들어진 각종 함수

  ✓ 일전에 우리는 함수에 대해 배웠고 함수를 만들어보았다.

  ✓ 이미 만들어져 있는 함수를 다시 만들 필요가 없기에 어느정도는 숙지하고 있는편이 프로그래밍에 도움이 된다.

# 07 파이썬 활용(내장함수)

## ▪ 내장함수

| 함수명/클래스명 | 기능 | 구분 |
|---|---|---|
| abs(x) | x의 절대값을 반환한다. | 내장함수 |
| all(iterable) | 모든 요소가 True 일 때 True를 반환한다. | 내장함수 |
| any(iterable) | 하나 이상의 요소가 True 일 때 True를 반환한다. | 내장함수 |
| bin(number) | 10진수 정수를 2진수로 반환한다. | 내장함수 |
| bool(x) | x를 논리형으로 반환한다. | 내장클래스 |
| complex(x) | x를 복소수형으로 반환한다. | 내장클래스 |
| dict(iterable) | 사전형 자료구조 형으로 반환한다. | 내장클래스 |
| dir(x) | 변수, 내장함수, 내장클래스의 목록을 반환한다. | 내장함수 |
| enumerate(iterable) | 열거형 자료를 순회하여 색인과 값을 반환한다. | 내장클래스 |
| eval(expr) | 문자열 수식을 계산 가능한 수식으로 반환한다. | 내장함수 |
| float(x) | 실수로 형 변환하여 반환한다. | 내장함수 |
| format(value) | value에 양식을 적용한다. | 내장함수 |
| help(x) | 함수 또는 클래스의 도움말을 반환한다. | 내장함수 |
| hex(number) | 10진수 정수를 16진수 값을 반환한다. | 내장함수 |
| id(obj) | 객체의 주소를 반환한다. | 내장함수 |
| input([prompt]) | 키보드로 입력한 문자열을 반환한다. | 내장함수 |
| int(x) | x를 정수형으로 반환한다. | 내장클래스 |
| len(obj) | 전체 원소의 길이를 반환한다. | 내장함수 |
| list((iterable) | list 자료구조로 형 변환한다. | 내장클래스 |
| max(iterable) | 최댓값을 반환한다. | 내장함수 |
| min(iterable) | 최솟값을 반환한다. | 내장함수 |
| oct(number) | 10진수 정수를 8진수 값으로 반환한다. | 내장함수 |
| open(file, mode) | 파일 읽기와 쓰기 | 내장함수 |
| ord(character) | character를 아스키 값으로 반환한다. | 내장함수 |
| pow(x, y) | x에 대한 y의 제곱을 계산하여 반환한다. | 내장함수 |
| print(value) | value를 콘솔에 출력한다. | 내장함수 |
| range(n) | 0에서 n-1 사이의 정수를 반환한다. | 내장클래스 |
| round(number) | 실수 number를 대상으로 반올림을 수행한다. | 내장함수 |
| set(iterable) | 셋 자료구조로 형 변환한다. | 내장클래스 |
| sorted(iterable) | 반복 가능한 원소들을 대상으로 정렬한다. | 내장함수 |
| str(object) | 문자형 자료형으로 변환한다. | 내장클래스 |
| sum(iterable) | 숫자들의 합을 구한다. | 내장함수 |
| tuple(iterable) | 튜플 자료구조로 형 변환한다. | 내장클래스 |
| type(x) | x의 자료형을 반환한다. | 내장클래스 |

# 07 파이썬 활용

## ▪ OS 모듈

- ✓ os(Operating System) 모듈은 운영체제에서 제공되는 여러 기능을 파이썬에서 수행할 수 있게 해준다.
- ✓ 폴더, 파일 생성 및 이동, 삭제 시 또는 디렉터리 내의 파일 목록 확인 시 사용

# 07 파이썬 활용

## ▪ OS 모듈

- ✓ 현재 작업중인 디렉토리의 경로를 보여주는 Current working dir
- ✓ print(os.getcwd())

- ✓ 현재 위치를 변경하는 chdir
- ✓ os.chdir('C:₩**path**')
- ✓ ₩n, ₩t 등 문자열 조작하는 이스케이프 문자 사용시 유의하라.

# 07 파이썬 활용

## ▪ OS 모듈

- ✓ 특정 디렉토리안에 있는 파일등의 목록을 보여주는 listdir
- ✓ print(os.listdir('C:₩**path**'))
- ✓ 폴더를 만드는 함수 mkdir (한번에 하나만)
- ✓ os.mkdir('C:₩chi_py_project₩₩test_dir')
- ✓ 비어있는 폴더 하나 삭제하는 함수 rmdir
- ✓ os.rmdir("C:₩chi_py_project₩₩test_dir")

# 07 파이썬 활용

- OS 모듈
  - ✓ 하위에 폴더를 연속적으로 만들어주는 makedirs 함수
  - ✓ os.makedirs('C:₩chi_py_project₩₩test_dir₩chi')
  - ✓ 비어있는 폴더를 하위까지 삭제하는 rmdir 함수
  - ✓ os.removedirs("C:₩chi_py_project₩₩test_dir₩chi")

# 07 파이썬 활용

- 파일 최초 생성하기
  - ✓ f = open(“새파일.txt”, ‘w’)
  - ✓ f.close()
  - ✓ r:읽기모드, w:등록모드, a:추가모드

# 07 파이썬 활용

- 생성된 파일 쓰기 모드로 열어 출력값 적기

  - ✓ f = open(“test.txt”, ‘w’)
  - ✓ for i in range(1,10):
  - ✓ data = “%d번째 줄입니다. ₩n” %i
  - ✓ f.write(data)
  - ✓ f.close()

# 07 파이썬 활용

- 생성된 파일 읽기 모드로 열기
  - ✓ f = open(“test.txt”, ‘r’)
  - ✓ line = f.readline()
  - ✓ print(line)
  - ✓ f.close()

# 07 파이썬 활용

- 생성된 파일 읽기 모드로 열기
  - ✓ f = open(“test.txt”, ‘r’)
  - ✓ lines = f.readlines()  >> lines는 list형태로 변수에 담김.
  - ✓ print(lines)
  - ✓ f.close()

# 07 파이썬 활용

- 생성된 파일 추가 모드로 열기

  - ✓ f = open(“test.txt”, ‘a’)
  - ✓ for i in range(10,21):
  - ✓ data = “%d번째 줄입니다. ₩n” %i
  - ✓ f.write(data)
  - ✓ f.close()

# 07 파이썬 활용

- 인코딩은 컴퓨터가 문자를 인식할 수 있도록 하는것
- 먼저 아스키 코드에 대해 알아보자.
- 아스키 코드는 일종에 컴퓨터와의 약속
- print(chr(97)), print(chr(98)) 등

# 07 파이썬 활용

- 아스키 코드

  ✓ 알파벳 포함 128개의 부호를 나타내는 숫자.

| Dec | Hx | Oct | Char | | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NUL | (null) | 32 | 20 | 040 |   | Space | 64 | 40 | 100 | @ | @ | 96 | 60 | 140 | ` | ` |
| 1 | 1 | 001 | SOH | (start of heading) | 33 | 21 | 041 | ! | ! | 65 | 41 | 101 | A | A | 97 | 61 | 141 | a | a |
| 2 | 2 | 002 | STX | (start of text) | 34 | 22 | 042 | " | " | 66 | 42 | 102 | B | B | 98 | 62 | 142 | b | b |
| 3 | 3 | 003 | ETX | (end of text) | 35 | 23 | 043 | # | # | 67 | 43 | 103 | C | C | 99 | 63 | 143 | c | c |
| 4 | 4 | 004 | EOT | (end of transmission) | 36 | 24 | 044 | $ | $ | 68 | 44 | 104 | D | D | 100 | 64 | 144 | d | d |
| 5 | 5 | 005 | ENQ | (enquiry) | 37 | 25 | 045 | % | % | 69 | 45 | 105 | E | E | 101 | 65 | 145 | e | e |
| 6 | 6 | 006 | ACK | (acknowledge) | 38 | 26 | 046 | & | & | 70 | 46 | 106 | F | F | 102 | 66 | 146 | f | f |
| 7 | 7 | 007 | BEL | (bell) | 39 | 27 | 047 | ' | ' | 71 | 47 | 107 | G | G | 103 | 67 | 147 | g | g |
| 8 | 8 | 010 | BS | (backspace) | 40 | 28 | 050 | ( | ( | 72 | 48 | 110 | H | H | 104 | 68 | 150 | h | h |
| 9 | 9 | 011 | TAB | (horizontal tab) | 41 | 29 | 051 | ) | ) | 73 | 49 | 111 | I | I | 105 | 69 | 151 | i | i |
| 10 | A | 012 | LF | (NL line feed, new line) | 42 | 2A | 052 | * | * | 74 | 4A | 112 | J | J | 106 | 6A | 152 | j | j |
| 11 | B | 013 | VT | (vertical tab) | 43 | 2B | 053 | + | + | 75 | 4B | 113 | K | K | 107 | 6B | 153 | k | k |
| 12 | C | 014 | FF | (NP form feed, new page) | 44 | 2C | 054 | , | , | 76 | 4C | 114 | L | L | 108 | 6C | 154 | l | l |
| 13 | D | 015 | CR | (carriage return) | 45 | 2D | 055 | - | - | 77 | 4D | 115 | M | M | 109 | 6D | 155 | m | m |
| 14 | E | 016 | SO | (shift out) | 46 | 2E | 056 | . | . | 78 | 4E | 116 | N | N | 110 | 6E | 156 | n | n |
| 15 | F | 017 | SI | (shift in) | 47 | 2F | 057 | / | / | 79 | 4F | 117 | O | O | 111 | 6F | 157 | o | o |
| 16 | 10 | 020 | DLE | (data link escape) | 48 | 30 | 060 | 0 | 0 | 80 | 50 | 120 | P | P | 112 | 70 | 160 | p | p |
| 17 | 11 | 021 | DC1 | (device control 1) | 49 | 31 | 061 | 1 | 1 | 81 | 51 | 121 | Q | Q | 113 | 71 | 161 | q | q |
| 18 | 12 | 022 | DC2 | (device control 2) | 50 | 32 | 062 | 2 | 2 | 82 | 52 | 122 | R | R | 114 | 72 | 162 | r | r |
| 19 | 13 | 023 | DC3 | (device control 3) | 51 | 33 | 063 | 3 | 3 | 83 | 53 | 123 | S | S | 115 | 73 | 163 | s | s |
| 20 | 14 | 024 | DC4 | (device control 4) | 52 | 34 | 064 | 4 | 4 | 84 | 54 | 124 | T | T | 116 | 74 | 164 | t | t |
| 21 | 15 | 025 | NAK | (negative acknowledge) | 53 | 35 | 065 | 5 | 5 | 85 | 55 | 125 | U | U | 117 | 75 | 165 | u | u |
| 22 | 16 | 026 | SYN | (synchronous idle) | 54 | 36 | 066 | 6 | 6 | 86 | 56 | 126 | V | V | 118 | 76 | 166 | v | v |
| 23 | 17 | 027 | ETB | (end of trans. block) | 55 | 37 | 067 | 7 | 7 | 87 | 57 | 127 | W | W | 119 | 77 | 167 | w | w |
| 24 | 18 | 030 | CAN | (cancel) | 56 | 38 | 070 | 8 | 8 | 88 | 58 | 130 | X | X | 120 | 78 | 170 | x | x |
| 25 | 19 | 031 | EM | (end of medium) | 57 | 39 | 071 | 9 | 9 | 89 | 59 | 131 | Y | Y | 121 | 79 | 171 | y | y |
| 26 | 1A | 032 | SUB | (substitute) | 58 | 3A | 072 | : | : | 90 | 5A | 132 | Z | Z | 122 | 7A | 172 | z | z |
| 27 | 1B | 033 | ESC | (escape) | 59 | 3B | 073 | ; | ; | 91 | 5B | 133 | [ | [ | 123 | 7B | 173 | { | { |
| 28 | 1C | 034 | FS | (file separator) | 60 | 3C | 074 | < | < | 92 | 5C | 134 | \ | \ | 124 | 7C | 174 | | | \| |
| 29 | 1D | 035 | GS | (group separator) | 61 | 3D | 075 | = | = | 93 | 5D | 135 | ] | ] | 125 | 7D | 175 | } | } |
| 30 | 1E | 036 | RS | (record separator) | 62 | 3E | 076 | > | > | 94 | 5E | 136 | ^ | ^ | 126 | 7E | 176 | ~ | ~ |
| 31 | 1F | 037 | US | (unit separator) | 63 | 3F | 077 | ? | ? | 95 | 5F | 137 | _ | _ | 127 | 7F | 177 | &#127; | DEL |

# 07 파이썬 활용

- MS949(Euc-kr)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| b0a0 | | 가 | 각 | 간 | 갇 | 갈 | 갉 | 갊 | 감 | 갑 | 값 | 갓 | 갔 | 강 | 갖 | 갗 |
| b0b0 | 같 | 갚 | 갛 | 개 | 객 | 갠 | 갤 | 갬 | 갭 | 갯 | 갰 | 갱 | 갸 | 갹 | 갼 | 걀 |
| b0c0 | 걋 | 걍 | 걔 | 걘 | 걜 | 거 | 걱 | 건 | 걷 | 걸 | 걺 | 검 | 겁 | 것 | 겄 | 겅 |
| b0d0 | 겆 | 겉 | 겊 | 겋 | 게 | 겐 | 겔 | 겜 | 겝 | 겟 | 겠 | 겡 | 겨 | 격 | 겪 | 견 |
| b0e0 | 겯 | 결 | 겸 | 겹 | 겻 | 겼 | 경 | 곁 | 계 | 곈 | 곌 | 곕 | 곗 | 고 | 곡 | 곤 |
| b0f0 | 곧 | 골 | 곪 | 곬 | 곯 | 곰 | 곱 | 곳 | 공 | 곶 | 과 | 곽 | 관 | 괄 | 괆 | |
| b1a0 | | 괌 | 괍 | 괏 | 광 | 괘 | 괜 | 괠 | 괩 | 괬 | 괭 | 괴 | 괵 | 괸 | 괼 | 굄 |
| b1b0 | 굅 | 굇 | 굉 | 교 | 굔 | 굘 | 굡 | 굣 | 구 | 국 | 군 | 굳 | 굴 | 굵 | 굶 | 굻 |
| b1c0 | 굼 | 굽 | 굿 | 궁 | 궂 | 궈 | 궉 | 권 | 궐 | 궜 | 궝 | 궤 | 궷 | 귀 | 귁 | 귄 |
| b1d0 | 궐 | 궘 | 귑 | 궷 | 규 | 균 | 귤 | 그 | 극 | 근 | 귿 | 글 | 긁 | 금 | 급 | 긋 |
| b1e0 | 긍 | 긔 | 기 | 긱 | 긴 | 긷 | 길 | 긺 | 김 | 깁 | 깃 | 깅 | 깆 | 깊 | 까 | 깍 |

# 07 파이썬 활용

- 유니코드(UTF-8)
  - ✓ 아스키 코드가 커버하지 못하는 한글, 한자 등 새로운 문자를 인코딩하기 위해 유니코드 등장
  - ✓ 유니코드 인코딩 방법중 하나인 UTF-8을 전세계적으로 가장 많이 사용
  - ✓ UTF-8 인코딩은 유니코드 한 문자를 나타내기 위해 1바이트에서 4바이트까지 가변적으로 사용됨

# 07 파이썬 활용

- 유니코드

# 07 파이썬 활용

- 인코딩/디코딩

  ✓ 파일에 입력된 한글이 왜 깨지는가?

  ✓ 인코딩시 한글을 인식할 수 있는 코드로 설정해주지 않았기 때문에

# 07 파이썬 활용

- HTML

  ✓ HTML(Hyper Text Markup Language)

  ✓ 웹페이지를 만들기 위한 목적으로 사용되며 웹브라우저 위에서 동작하는 언어

  ✓ 태그로 이루어져 있으며 정적인 정보를 담는다

  ✓ 실습을 통해 각종 태그들이 어떤 역할을 하는지 알아본다

# 07 파이썬 활용

```
<html>
<head>
    <meta charset="utf-8">
    <title>타이틀</title>
</head>
</head>

<body>
    <h1>hn태그</h1>
    <p>P태그</p>
    <hr>
    <a href="사이트 주소">a 태그</a>
    <img src="이미지 주소" width="500" heigth="200">
</body>
</html>
```

# 07 파이썬 활용(정규표현식)

- 정규식
  - ✓ 정규식의 개념
  - ✓ 정규식을 활용하는 다양한 방법

# 07 파이썬 활용(정규표현식)

- 정규표현식
  - ✓ 웹크롤링, 데이터 분석시 특정한 데이터, 특정한 구문, 패턴등을 골라내는 작업이 필요하다. 이때 사용되는것이 정규 표현식
  - ✓ import re
  - ✓ 복잡한 데이터 내 특정부분만 추출하고자 할때 사용

# 07 파이썬 활용(정규표현식)

| 문법 | 설명 |
| --- | --- |
| . | 임의의 한 문자가 존재 |
| ? | 바로 앞의 문자가 존재하거나 존재하지 않음 |
| * | 바로 앞의 문자가 존재하지 않거나<br>무한대로 존재 |
| + | 바로 앞의 문자가 한 번 이상 존재 |
| ^ | 바로 뒤의 문자로 문자열이 시작 |
| $ | 바로 앞의 문자로 문자열이 끝남 |

# 07 파이썬 활용(정규표현식)

| 문법 | 설명 |
| --- | --- |
| {숫자} | 숫자만큼 반복 |
| {숫자,} | 숫자 이상만큼 반복 |
| {숫자1,숫자2} | 숫자 1 이상, 숫자 2 이하만큼 반복 |
| (문자열) | 문자나 문자열을 묶음 |
| [문자1,문자2...] | 대괄호 안에 있는 문자들이 존재하는지 검색 |
| [ ^ ] | '^' 기호 바로 뒤에 문자가 존재하지 않음 |

# 07 파이썬 활용(정규표현식)

| 문법 | 설명 |
| --- | --- |
| ₩₩ | ₩ (역슬래쉬) 글자 자체를 검색 |
| ₩d | 모든 숫자를 검색 . [0-9] 와 동일 |
| ₩D | 숫자를 제외한 모든 문자를 검색 |
| ₩s | 공백을 검색 |
| ₩S | 공백이 아닌 문자를 검색 |
| ₩w | 숫자 또는 문자를 검색 [a-zA-Z0-9] |
| ₩W | 숫자 또는 문자가 아닌 것을 검색 |