

인코딩 디코딩의 기본개념을 잡아야 겠다.

encode를 하여 cp949, utf-8, euc-kr등으로 할 수 있다.

encode를 하면 type이 byte로 된다.

decode는 현재 기본적으로 쓰이는 coding방식을 사용하여야 한다. 나 같은 경우는 cp949이다.

python 같은 경우는 # -- coding: cp949 -- 로 coding방식을 정해줄 수 있다.

그냥 %c를 출력하면 ascii코드 값에 해당하는 값이 출력 된다.

b\글자코드\글자코드를 decode해줘야한다

```
In [12]: # -*- coding: cp949 -*-
import sys
sys.stdin.encoding
|
a = "가"
a_number = a.encode('cp949')
print(a.encode('cp949'))
print(type(a_number))
print(a_number.decode('cp949'))
print("%c" % (0x801))
print(b'\xb0\xa1'.decode('cp949'))

b = "어렵다"
b_bytes = b.encode('cp949')
print(b_bytes)
print(type(b_bytes))
print(b_bytes.decode('cp949'))

test_bytes = bytes([0xa1])
test_bytes2 = bytes([0xb2])
print(test_bytes + test_bytes2)

b'\xb0\xa1'
<class 'bytes'>
가
날
가
가
b'\xb0\xff\xe0\xff\xb7\xff\xc6\xff\xb4\xff\xd9'
<class 'bytes'>
어렵다
b'\xa1\xff\xb2'
```

1.1 [A1~FE][A1~FE] 영역 전체 코드표를 출력하시오.

```

In [17]: for i in range(0xA1, 0xFF):
          for j in range(0xA1, 0xFF):
              h_bytes = bytes([i] + bytes([j]))
              # print(h_bytes)
              hcode = (i * 256) | j
              try:
                  print("%c : 0x%x(%d)  %s(h_bytes.decode('cp849'), hcode, hcode))"
              except:
                  print("??? : ", h_bytes)

```

```

: 0x01a1(41377)
: 0x01a2(41378)
: 0x01a3(41379)
: 0x01a4(41380)
: 0x01a5(41381)
: 0x01a6(41382)
: 0x01a7(41383)
: 0x01a8(41384)
: 0x01a9(41385)
- : 0x01aa(41386)
|| : 0x01ab(41387)
: 0x01ac(41388)
~ : 0x01ad(41389)
: 0x01ae(41390)
: 0x01af(41391)
: 0x01b0(41392)
: 0x01b1(41393)
[ : 0x01b2(41394)
: 0x01b3(41395)
< : 0x01b4(41396)

```

1.2 한글 2,350자(B0~C8)만 출력하시오.

```
In [13]: for i in range(0x00A1, 0x08FF):
          a = i.to_bytes(2, 'big')
          try:
              print(a.decode('cp949'))
          except:
              print("?? : bytes ", a)
          a.replace('00', "HW", 1)
          print(a)
```

가
각
간
갈
감
갑
갸
갯
강
갇
갸
갈
갸
갈
개
개

1.3 한자 4,888자(CA~FD)만 출력하시오.

```
In [16]: for i in range(0xCAAF, 0xFEFF):
a = i.to_bytes(2, 'big')
try:
    print(a.decode('cp949'))
except:
    print("?? : bytes ", a)
```

伽佳假價加可呵哥嘉嫁家暇架枷柯歌珂痂稼

1.4 CP949의 8,822자를 출력하시오

```
[7]: for i in range(0x81, 0xA1):
      for j in range(0x41, 0x5B):
          h_bytes = bytes([i]) + bytes([j])
          try:
              print(h_bytes.decode('cp849'))
          except:
              print("?? : bytes ", h_bytes)

      for j in range(0x61, 0x7B):
          h_bytes = bytes([i]) + bytes([j])
          try:
              print(h_bytes.decode('cp849'))
          except:
              print("?? : bytes ", h_bytes)

      for j in range(0x81, 0xFF):
          h_bytes = bytes([i]) + bytes([j])
          try:
              print(h_bytes.decode('cp849'))
          except:
              print("?? : bytes ", h_bytes)

for i in range(0xA1, 0xC6):
    for j in range(0x61, 0x7B):
        h_bytes = bytes([i]) + bytes([j])
        try:
            print(h_bytes.decode('cp849'))
        except:
            print("?? : bytes ", h_bytes)

    for j in range(0x61, 0x7B):
        h_bytes = bytes([i]) + bytes([j])
        try:
            print(h_bytes.decode('cp849'))
        except:
            print("?? : bytes ", h_bytes)
```

```
for i in range(0x41, 0x52):
    h_bytes = bytes([0x06]) + bytes([i])
    try:
        print(h_bytes.decode('cp849'))
    except:
        print("?? : bytes ", h_bytes)
```

[81-A0][41-5A, 61-7A, 81-FE]

[A1-C5][41-5A,61-7A,81-A0]

C6[41-52]

[illegible]

2. 임의의 유니코드 값에 대한 한글 코드값(10진수 또는 16진수)과 <초성, 중성, 종성>값을 출력하시오.

```
In [89]: # -*- coding: utf-8 -*-
import random
rand_num = random.randint(1, 11173) % 11172 + 0xAC00

# test = 0xd7a9
chosung = int(((rand_num-0xAC00) / 28) / 21) % 19
joongsung = int(((rand_num-0xAC00) / 28) % 21)
jongsung = int((rand_num-0xAC00) % 28)
print("글자 : {}, 숫자 : {}, - (초성: {}, 중성: {}, 종성: {})".format(chr(rand_num), hex(rand_num), chosung, joongsung, jongsung))

# 0xAC00 -> {0, 0, 0}
# 0xAC01 -> {0, 0, 1}
# 0xD7A9 -> {19, 20, 27}
# □ 초성
# ( (# 코드값-0xAC00) / 28 ) / 21 ) % 19
# □ 중성
# (
# 코드값-0xAC00 ) / 28 ) % 21
# □ 종성
# (
# 코드값-0xAC00 ) % 28

글자 : 님, 숫자 : 0xb2d5, - (초성: 2, 중성: 20, 종성: 13 )
```

3. 임의의 유니코드 음절을 자모를 분리하여 출력하시오.

```
In [90]: # - UTF-8 인코딩 방법

# 0xxx xxxx // 7 비트 영역
# 110x xxxx 10xxxxxx // 8~11 비트 영역
# 1110 xxxx 10xxxxxx 10xxxxxx // 12~16 비트 영역
# 1111 0xxx 10zzxxxx 10xxxxxx 10xxxxxx // 17~21 비트 영역
print("초성: ", chr(0x1100 + chosung))
print("중성: ", chr(0x1161 + joongsung))
print("종성: ", chr(0x11a7 + jongsung))

# 초성: U+1100~U+1112, 중성: U+1161~U+1175, 종성: U+11A8~U+11C2

초성 : ㄴ
중성 : ㅣ
종성 : ㄹ
```

* 알게된 점

python에서 바이트형으로 변환후 더하면 2byte 되는 것

encoding 방식 설정해줄수 있는 것

encode decode 방식 설정해 줄수 있는 것 ex) euc-kr, cp949, utf-8, ascii

utf-8은 입력마스크에 의해 다시 인코딩 되는데 초성 중성 종성값을 따로 받아서 출력해 줄 수 있다.