

[문자열 9]

```
1 # p020.py
2 a = "나는 %d살" % 20
3 age = 21
4 b = "나는 %d살" % age
5 c = "내 이름 %s" % "인하"
6 name = "정석"
7 d = "내 이름 %s" % name
8 e = "이름:%s 나이:%d" % (name, age)
9
10 print(a)
11 print(b)
12 print(c)
13 print(d)
14 print(e)
```

[문자열 10]

```
1 # p021.py
2 a = "%d년 %d월 %d일 %s 온도는 %f도 입니다" % (2021, 9, 1, "최저", 20.5)
3 b = "%d년 %d월 %d일 %s 온도는 %f도 입니다" % (2021, 9, 1, "최고", 20.5)
4 c = "%d년 %d월 %d일 %s 온도는 %.1f도 입니다" % (2021, 9, 2, "최저", 20.5)
5 print(a)
6 print(b)
7 print(c)
8
9 s1 = "A%10sZ" % "az"
10 s2 = "A%-10sZ" % "az"
11 print(s1)
12 print(s2)
13
14 f1 = "%0.5f" % 3.141592
15 f2 = "%10.5f" % 3.141592
16 print(f1)
17 print(f2)
```

[문자열 11]

```
1 # p022.py
2 a = "나는 {0}개의 사과를 먹을래요!".format(3)
3 b = "나는 {0}개의 사과를 먹을래요!".format("세")
4 print(a)
5 print(b)
6
7 number = 3
8 days = "삼일"
9 c = "나는 {0}개의 사과를 먹었고, 그래서 {1}동안 아팠어요".format(number, day)
10
11 number = 3
12 days = "삼일"
13 c = "나는 {0}개의 사과를 먹었고, 그래서 {1}동안 아팠어요".format(number, days)
14 print(c)
15
16 d = "나는 {num}개의 사과를 먹었고, 그래서 {day}동안 아팠어요".format(num=3, day="삼일")
17 print(d)
```

[문자열 12]

```
1 # p023.py
2 a = "'{0:<10}'".format("hi") # 왼쪽정렬
3 print(a) # (''hi', 'hi')
4 a = "'{0:<10}'".format("hi") # 왼쪽정렬
5 print(a) # 'hi'
6 a = "'{0:>10}'".format("hi") # 오른쪽정렬
7 print(a) # 'hi'
8 a = "'{0:^10}'".format("hi") # 가운데정렬
9 print(a) # 'hi'
10 a = "'{0:=^10}'".format("hi")
11 print(a) # '===hi==='
12 a = "'{0:!<10'".format("hi")
13 print(a) # 'hi!!!!!!!'
14
15 y = 3.141592
16 s = "{0:0.4f}".format(y)
17 print(s) # 3.1416
```

[문자열 13]

```
1 # p024.py
2 name = "김인하"
3 age = 20
4 intro = "내 이름은 {name}입니다. 나이는 {age}살 입니다."
5 print(intro)
6
7 intro = f"내 이름은 {name}입니다. 나이는 {age}살 입니다."
8 print(intro)
9
10 intro = f"내 이름은 {name}입니다. 나이는 {age+1}살 입니다."
11 print(intro)
```

[문자열 14]

```
1 # p025.py
2 a = "hi"
3 left = f"|{'hi':<10}|" # 왼쪽 정렬
4 center = f"|{a:^10}|" # 가운데 정렬
5 right = f"|{a:>10}|" # 오른쪽 정렬
6 print(left)
7 print(center)
8 print(right)
9
10 center2 = f"|{a:=^10}|" # 가운데 정렬 + 빈 공백은 문자로 대체
11 right2 = f"|{a:=>10}|" # 오른쪽 정렬 + 빈 공백은 문자로 대체
12 print(center2)
13 print(right2)
14
15 pi = 3.141592
16 str_pi = f"{pi:0.2f}" # 소수점 두 자리만 표기
17 print(str_pi)
18
19 str_pi = f"{pi:10.2f}" # 10자리로 소수점 두 자리만 표기
20 print(str_pi)
21
22 str_pi = f"{pi:010.2f}" # 10자리로 소수점 두 자리만 표기 + 빈 공백은 0으로 대체
23 print(str_pi)
```

[문자열 15]

```
1 # p026.py
2 test = "i am a BOY."
3 print(test.count('a'))
4
5 print(test.find('a'))
6 print(test.find('q'))
7
8 print(test.index('a'))
9 print(test.index('q'))
10
11 print("/".join(test))
12 print(test.upper())
13 print(test.lower())
14 print(test.title())
15 print(test.capitalize())
```

[문자열 16]

```
1 # p027.py
2 test = " JMT University "
3
4 print("|" + test.strip() + "|")
5 print("|" + test.lstrip() + "|")
6 print("|" + test.rstrip() + "|")
7
8 print(test.replace("University", "High School"))
9
10 print(test.split()) # 리스트로 반환된다.
11 print(test.split('i')) # 리스트로 반환된다.
12
13 print(test)
```

[List 의 순서 변경]

```
1 # p036.py
2 cars = ['audi', 'tesla', 'benz', 'kia', 'lincoln', 'hyundai']
3 print(cars)
4
5 # 오름차순(정방향, asc)으로 정렬
6 cars.sort()
7 print(cars)
8
9 # 내림차순(역방향, desc)으로 정렬
10 cars.sort(reverse=True)
11 print(cars)
12 print("-" * 60)
13 cars = ['audi', 'tesla', 'benz', 'kia', 'lincoln', 'hyundai']
14
15 # 오름차순(정방향, asc)으로 정렬
16 cars_copy = sorted(cars)
17 print(cars_copy)
18
19 # 내림차순(역방향, desc)으로 정렬
20 cars_copy = sorted(cars, reverse=True)
21 print(cars_copy)
22 print(cars)
23
24 print("-" * 60)
25 cars = ['audi', 'tesla', 'benz', 'kia', 'lincoln', 'hyundai']
26 cars.reverse()
27 print(cars)
28
29 print("-" * 60)
30 # 원본 손상없이 reverse()를 하려면?
31 cars = ['audi', 'tesla', 'benz', 'kia', 'lincoln', 'hyundai']
32 cars_copy = cars[:] # 복사
33 cars_copy.reverse()
34 print(cars)
35 print(cars_copy)
```

[List 와 if]

```
1  # p046.py
2  req_topping = ["버섯", "양파", "파인애플", "페퍼로니"]
3
4  for topping in req_topping:
5      print(f"{topping} 추가")
6
7  print("피자 완성!")
8
9  print("*" * 30)
10 req_topping = ["버섯", "양파", "파인애플", "페퍼로니"]
11
12 for topping in req_topping:
13     if topping == "페퍼로니":
14         print(f"페퍼로니 재고가 없어요")
15     else:
16         print(f"{topping} 추가")
17
18 print("피자 완성!")
19
20 print("*" * 30)
21 req_topping = []
22
23 if req_topping:
24     for topping in req_topping:
25         if topping == "페퍼로니":
26             print(f"페퍼로니 재고가 없어요")
27         else:
28             print(f"{topping} 추가")
29     print("피자 완성!")
30 else:
31     print("모든 재고 소진!")
32
33 print("*" * 30)
34 #가게 보유 토핑 재고
35 avl_topping = ["버섯", "피망", "치즈", "올리브", "양파", "페퍼로니"]
36 #요구 토핑
37 req_topping = ["버섯", "양파", "파인애플", "페퍼로니"]
38
39 for topping in req_topping:
40     if topping in avl_topping:
41         print(f"{topping} 추가")
42     else:
43         print(f"{topping} 재고가 없어요 ㅜ_ㅜ")
44
45 print("피자 완성!")
```

[for loop]

```
1  # p050.py
2  scores = []  #list()
3
4  data = input("2번:")
5  data = int(data)
6  scores.append(data)
7
8  scores.append(int(input("3번:")))
9  scores.append(int(input("4번:")))
10 scores.append(int(input("5번:")))
11 scores.insert(0, (int(input("1번:"))))
12
13 number = 0
14 summary = 0
15 for score in scores:
16     number += 1  # number = number + 1
17     summary += score  # summary = summary + score
18     print(f"{number}번 학생 점수:", score)
19
20 avg = summary / len(scores)
21 print("총 평균:", avg)
```

[for Loop 와 list]

```
1  # p053.py
2
3  req_topping = ["버섯", "양파", "파인애플", "페퍼로니"]
4  for topping in req_topping:
5      print(f"{topping} 추가")
6  print("피자 완성!")
7
8  req_topping = ["버섯", "양파", "파인애플", "페퍼로니"]
9  for topping in req_topping:
10     if topping == "페퍼로니":
11         print(f"{topping} 재고가 없어요")
12     else:
13         print(f"{topping} 추가")
14  print("피자 완성!")
15
16  req_topping = []
17  if req_topping:
18      for topping in req_topping:
19          if topping == "페퍼로니":
20              print(f"{topping} 재고가 없어요")
21          else:
22              print(f"{topping} 추가")
23      print("피자 완성!")
24  else:
25      print("모든 재고 소진!")
26
27  print("***30)
28  # 가게 보유 토핑 재고
29  avl_topping = ["버섯", "피망", "치즈", "올리브", "양파", "페퍼로니"]
30  # 요구 토핑
31  req_topping = ["버섯", "양파", "파인애플", "페퍼로니"]
32
33  for topping in req_topping:
34      if topping in avl_topping:
35          print(f"{topping} 추가")
36      else:
37          print(f"{topping} 재고가 없어요 T_T")
38  print("피자 완성!")
```