

# 함수와 논리흐름 제어

## 함수

- 스크립트 : 여러 명령어를 하나의 스크립트로 만들어 일련의 작업을 자동화
  - 파일로 저장가능, 실행 가능
  - 스크립트에 포함된 데이터가 아닌경우 스크립트를 고쳐야 함(가변적인 입력 데이터를 처리할 때 유연하지 못함)
- source() 함수 : 저장된 스크립트 파일 호출
- 함수 : 연속적으로 수행되는 일련의 코드 집합, 스크립트와 크게 다르지않음
  - 가변적인 입력 데이터 처리 가능, input에 따라 output이 다라짐
  - 실행결과를 객체로 변환 할 수 있음
    - 실행 결과를 다른 작업에서 쉽게 이용할 수 있음
- function() 함수 : 사용자 정의 함수 생성
  - 괄호 안에는 함수 내에서 사용할 인수 지정
    - 여러개 지정 가능
  - 실행할 코드는 중괄호 안에 포함
    - 꼭 필요한 건 아님(한줄로 작성 가능), 가독성이 떨어짐
  - return() 명령어는 정의된 함수의 실행 결과를 함수 내부에서 작업공간으로 보내는 역할, 객체 하나만 반환
    - 꼭 필요한건 아님, 코드의 끝까지 실행하지 않고 주약녕 함수를 벗어나고 싶을 때 필요
  - 함수의 정의가 완료되면 <-를 이용해 함수에 이름 부여
  - 함수를 사용하기 위해서는 함수를 실행해 메모리(작업공간)에 적재
  - 다른 유형의 객체들과 동일하게 취급 -> 복사 가능
    - 동일한 이름의 다른 객체가 존재하면 기존의 객체를 메모리에서 삭제하니 주의
- 인수
  - 함수를 정의할 때 사전에 지정한 값을 인수에 할당 가능, 호출 시 인수 지정을 하지 않으면 이 값 사용
  - 인수는 이름을 가져야 함, 호출 시 정의된이름 사용, 이름이 없다면 순서대로 인수로 취급
  - 가변적 인수 리스트 포함 가능
    - ... : 내부 함수에서 사용하게 될 인수를 총칭, 지정하지 않으면 해당 내부함수에 지정된 초기값 사용
  - 함수 자체를 인수로 지정할 수 있음
    - 인수 리스트에 FUN 인수 추가, 원하는 함수 호출, 초기값 지정 가능

## 논리흐름 제어

- if 문 : 괄호 안의 조건을 평가해 그 값이 TRUE이면 다음 문장 수행
  - if else 문도 가능
    - 벡터 연산은 수행 불가능(if 다음 괄호 안에는 하나의 논리값만이 존재)
    - 첫번째 값만이 논리 연산을 위해 사용

- ifelse() 함수 : 조건문에 벡터를 포함시켜 벡터 연산 하기
  - 첫번째 인수 : 조건
  - 두번째 인수 : TRUE시 실행할 것
  - 세번째 인수 : FALSE시 실행할 것
- switch() 함수
  - 첫번째 인수 : 값
  - 두번째 인수 : 값에 따라 실행할 코드(값 = 코드 형식)
- repeat 문 : 반복 작업 수행 명령어
  - break 명령어 : 반복 작업 멈추기
  - next 명령어 : 반복 작업 도중 다음단계로 건너뛰기
- while 문 : 조건이 TRUE이면 작업 반복 수행
  - break, next 사용 가능
- for 문 : 벡터 내의 각 원소에 대해 차례대로 반복 작업 수행
  - for (*var* in *list*) *statement*
  - *list* 내의 각 원소를 차례대로 추출해 *var*에 할당한 후 *statement* 코드 반복 수행
  - break, next 사용 가능
  - 실행 결과는 루프 내에서 출력되지 않음 : 명시적으로 print() 함수 호출 필요
  - 함수와 달리 for 루프 내에서 생성된 객체는 for 문이 종료된 뒤에도 사라지지 않음