Chapter 2

스칼라 언어로 코딩하기 앞서 스칼라 프로그램에 대해 살펴본다.

Step 1. 스칼라 인터프리터 배우기

인터프리터는 스칼라 표현식과 프로그램을 작성할 수 있는 대화형 셸이다. 단순한 표현식을 입력하면 결과 값을 출력한다. 명령 프롬프트에서 scala를 입력하여 사용할 수 있는데 1+2를 입력하면 res0: Int = 3 이라는 문장이 출력된다. Println 함수로 원하는 문장을 출력할 수도 있다.

Step 2. 변수를 정의하라

스칼라에는 두 종류의 변수가 있는데 val은 한번 초기화되면 다시는 값을 재할당할 수 없다. 반면 var는 재할당이 가능하다.

Step 3. 함수를 정의해라

함수 정의는 def로 시작한다. 함수이름 뒤에 괄호로 묶인 매개 변수 목록이 쉼표로 구분되어 나온다. 함수 매개변수에는 콜론 뒤에 타입 주석이 필요하다 (Int, String). 또한 함수 자체의 반환 타입을 정의한다. 이렇게 정의한 함수는 함수 이름과 매개변수로 호출할 수 있다. 만약 함수가 특별한 값을 반환하지 않는다면 ()Unit이 출력된다.

Step 4. 스칼라 스크립트 작성하기

스칼라에는 args라는 스칼라 배열을 통해 인수를 받아 사용할 수 있다. 배열은 0부터 시작하며 첫 번째 요소는 steps(0)로 접근한다. 이를 통해 사용자가 원하는 인수를 제공하면 그 값을 활용할 수 있다.

Step 5. 반복문 활용하기

while문을 사용하면 인덱스를 사용하여 인덱스 값을 조절하며 조건에 맞을 때까지 반복하는 C언어와 비슷한 기능을한다.

Step 6. For 반복문

for문과 foreach문을 사용하면 index 없이 반복문을 사용할 수 있다. for (arg <- args) println(arg)

이런식으로 args로 인수를 받아서 for문을 이용해 출력하면 한 줄에 한 단어씩 출력된다.

Step 7. 배열에 타입을 매개변수로 전달하기

New 키워드를 통해 객체, 즉 클래스를 생성할 수 있다. 또한 배열의 인덱스는 괄호를 통해 접근 가능하면 apply 메서드 호출로 변환된다. Val로 선언된 변수는 재할당할 수 없지만 그 변수가 참조하는 객체의 요소는 변경할 수 있다.

Step 8. 리스트 사용하기

함수형 프로그래밍에서는 값을 계산해 변환하는 데 집중하여 객체를 불변하게 만드는 것이 중요하다. 스칼라에서 array는 가변이며 list는 불변이다. 리스트는 다음과 같이 생성할 수 있다: val oneTwoThree= List(1,2,3). 리스트는 ::: 매서드를 사용하여 리스트간 병합이 가능하다. :: 매서드를 사용하면 리스트 앞에 요소를 추가할 수 있고 빈 리스트를 표현하는 Nil을 사용하면 리스트를 초기화할 수 있다. 리스트는 head, tail 같은 다양한 명령어가 있다.

Step 9. 튜플 사용하기

튜플은 불변하는 객체이다. 리스트와 달리 다양한 타입의 요소를 함께 담을 수 있다. 튜플을 사용할 때는 요소를 괄호 안에 쉼표로 구분하여 넣고, _1 _2등의 방법으로 접근한다. 튜플은 포함된 요소의 수와 각 요소의 타입에 따라 결정된다. Ex) Tuple_name[Int, String]

Step 10. Set, map 사용하기

Set과 Map은 계층 구조로 모델링 되어있다. 집합은 불변집합과 가변집합으로 나뉘는데 불변집합은 요소를 추가할 때 새로운 집합을 반환한다. 기본적으로 불변집합이 생성된다. 가변집합은 집합에 직접 요소를 추가하거나 제거할 수 있지만 추가적인 패키지를 import 해야한다. Map도 가변 Map과 불변 map으로 나뉜다. 가변 map은 패키지를 사용하여 생성할 수 있으며 키와 값을 추가하거나 변경할 수 있다.

Step 11. 함수형 스타일 인식하기

함수형 스타일과 명령형 스타일 구분하기

- 변수와 상수: 코드에 var과 val을 적절히 사용하여 함수를 작성하자.

함수형 프로그래밍의 장점

- 코드가 더 명확하고 읽기 쉬우며 오류 발생 가능성을 줄인다.
- 부수 효과를 최소화할 수 있다.

Step 12. 파일 읽기

- 파일 읽기 및 출력: scala.io.Source의 Source 클래스를 사용하여 파일을 읽는다.
- 출력 형식 조정: 파일을 list로 변환하여 사용할 수 있다.
- 형식에 맞춰 출력: 각 줄의 길이 앞에 공백을 추가하여 최대 너비를 맞춘다.