

Mingoonscript 쉬운 언어 명세서

이 명세서는 명세와 설명의 역할 모두를 위해 작성되었으며 언어 구현을 위해 보다 자세한 명세를 얻으려면 다른 문서를 참조하십시오.

데이터 타입

Mingoonscript 은 동적 타입 지정 언어(dynamic typed language) 이며 적은 종류의 데이터 타입만이 존재한다. 또한 모든 데이터 타입은 일등급 값으로 취급된다.

Capsule (캡슐)

캡슐은 하나의 데이터 타입이며 다른 언어에서 변수, 배열, 함수등으로 불리우는 다양한 객체들 역시 모두 캡슐로 취급한다. 다만 담긴 내용에 따라 역할만 달라질 뿐이다. 즉 모든 것은 캡슐(다른 언어- ruby 등에서 보자면 모든 것은 객체)이며 단지 그 안에 내포하는 내용물의 종류에 따라 하는 일이 달라지는 것이다.(루틴을 담으면 함수, 값을 담으면 변수 정도로 생각하면 된다)

캡슐의 크기에는 제한이 없다. 즉 언어 명세에서는 제한을 두지 않으나, 언어 구현체와 실행 환경에 따라 달라진다.

앞으로 이하 설명에서는 이해를 돕기 위해 캡슐 앞에 별도의 호칭을 붙여 설명할 것이다.

Nil (없음 값)

Nil 은 주로 다른 타입과 다르다는 것을 나타내는 목적으로 사용되며 단 하나의 값인 nil 만을 가질 수 있는 데이터 타입이다. 이 값은 유효하는 값이 존재하지 않음을 나타내기 위해 사용된다.

Number (수치)

수치 타입은 배정도 부동소수점(double-precision floating point) 수치를 뜻하며 정수 값이 아닌 실수 값을 뜻한다.

수치 상수를 표현할때는 소수점 이하 부분을 지정할 수 있으며 지수도 지정이 가능하다.

String (문자열)

String 은 유니코드 UTF-8 인코딩을 기본으로 표현되고 또한 그렇게 표현되고 있다고 간주되는 데이터 타입이다. 그 값들의 앞뒤에 쌍따옴표(") 를 사용하여 감싸준다. 또한 탈출열을 사용 할 수도 있다. 사용할 수 있는 탈출열의 종류에는 다음과 같은 것들이 있다.

\n 줄바꿈

\” 큰따옴표
\t 수평탭
\n 백슬래시

이 탈출열들을 그냥 그대로 표현하려면 다른 언어에서 그랬던 것처럼 앞에 \ 을 또 붙여주면 된다.

키워드

() =

연산자

사칙 연산자

덧셈 : $a + b$

뺄셈 : $a - b$

곱셈 : $a * b$

나눗셈(몫) : a / b

대입 연산자 $a = b$

비교 연산자 $> < >= <= != ==$

캡슐 - 네이밍 규칙

키워드는 캡슐의 이름으로 사용할 수 없다.

캡슐 - 변수와 배열의 역할

C에서 변수를 사용하려면 우선 선언하고, 그 다음 값을 지정한다.

예를 들면 `int var; var = 1;` 이렇게 되는데, Mingoonscript는 동적 타입 언어기 때문에 그냥 필요 할 때 그 이름을 명시해 주면 된다.

값을 담은 캡슐은 자동으로 그 값의 데이터 타입을 추론하여 작동한다.(동적 타입 및 타입 자동 추론 시스템)

또한 배열처럼 key 를 주어 값을 취급하려면 다음과 같이 사용하면 된다.

`Capsule_name(key)`

Key 는 String Data Type 으로도 지정이 가능하다.

변수와 배열의 값을 호출할때는 그 변수나 배열의 이름을 써 주면 되고, 값을 지정할 때도 간단하게 이름 = 값 형식을 취하면 된다.

캡슐 - 함수의 역할

루틴을 담는 캡슐은 다른 언어에서 보면 일종의 함수라 볼 수 있다.

함수 호출법은 다음과 같다.

Function(argument)

만약 인자가 여러 개면 쉼표(.)로 서로를 구분해주면 되고 인자가 없으면 그냥 ()로 냅두면 된다.

함수를 정의할 때는 다음과 같이 하면 된다.

```
FunctionExam = (  
    내용  
)
```

함수에서 자신에게 넘어온 인자를 처리하려면 인자를 가져오는 함수를 사용하면 된다. Mingoonscript Standard Library 에서는 그런 함수를 다음과 같이 정의한다.(자세한 것은 표준 라이브러리 문서를 참고하기를 바란다)

getArg(what)

만약 f(x,y) 와 같이 호출되었다면 getArg(1) 은 x 고 getArg(2) 는 y 가 되는것이다. 다른 언어에서 사용되는 return 기능은 다음과 같이 사용한다.

return value

Mingoonscript 에서는 여러 값을 동시에 return 할 수 있다.

이와 관련해서는 다음 코드를 참조하도록 한다.(다음 코드는 정확한 문법이 아니며 설명을 위해 임의로 사용하는 문법이다)

```
f = ( return 1, 2, 3 )  
x = f()  
    >>> 1  
y, z = f()  
    >>> y:1 z:2  
xx, yy, zz = f()  
    >>> xx:1 yy:2 zz:3
```

즉 one by one 방식을 통해 return 하며 서로 개수가 맞지 않을때는 return 값을 잘라서 전달한다.

제어 구조 문법

Mingoonscript 에는 여러 제어 구조가 존재한다.

조건문 if

```
if 조건 (
    조건이 맞을 때 실행
)
```

해당 조건을 평가해 유효하다면 루틴을 실행한다.

그 외의 경우에 대비해 else 도 사용 할 수 있다.

```
if 조건 (
    조건이 맞을 때 실행
)
Else (
    예외경우
)
```

반복문 loop

```
loop (
    실행
)
```

loop 를 종료하려면 루틴을 종료하는 함수를 호출하면 되며 표준 라이브러리에서는 이 함수를 end() 로 정의하고 있습니다.