

Wops Grammar Reference v1.0

실행되는 의사코드를 향해

이 프로그래밍 언어는 '실행되는 의사코드'라는 목적 하에 PS, Algorithm을 개발하는 데 사용될 예정입니다. 그 말은 즉슨 실무 사용에는 관계가 없게 할 예정입니다. 다음은 Wops 개발에 참여하기 전 참고해야 하는 점을 모아봤습니다.

- 이 프로그래밍 언어는 '단순한 문법'을 지향합니다. Golang처럼 필요한 문법만 넣을 예정입니다. 그렇기 때문에 문법 제안 전 합당한 이유를 준비해주세요
- 아무래도 프로그래밍 언어이기 때문에 어느정도 효율성은 따질 것입니다. 메모리 효율, 속도 효율을 잘 따져주세요.
- 자료구조 추가는 언제나 환영입니다!

Index

- [Data Type](#)
- [Lvalue](#)
- [Control Expression](#)
- [Function](#)
- [Structure](#)
- [Miscellaneous](#)

Data Type

<type>에 사용되는 데이터 타입으로, vector과 같은 컨테이너와는 다른 개념입니다.

List

- int: 정수 데이터 타입. v0.1에서는 C++ signed int으로 대체되지만 추후에 범위를 자동으로 조절해주는 시스템을 개발할 것입니다.
- double: 부동소수점 타입. 이 역시 v0.1에서는 C++ signed double로 대체됩니다.
- bool: 부울 타입. C++ bool로 고정되며 업데이트 계획은 없습니다.
- string: 문자열 타입. C++ std::string으로 고정되며 정식 출시 시 문자열 관련 함수를 추가할 예정입니다.

Lvalue

Token Description

<type>: 자료형

<name>: 사용되는 모든 문자열 토큰 (문자 + (0문자 이상의 문자열 & 0개 이상의 숫자))

<init_value>와 <value>는 value 가능

Grammar Expression

- initialize

```
<type> <name> = <init_value>
```

- constant value

```
const <type> <name> = <value>
```

- modify lvalue

```
<name> = <value>
```

Control Expression

Token Description

<statement>: 부울값을 반환하는 모든 식 & value

<variable>: 임시 변수 이름

<start>, <end>, <step>: range 함수 argv

<list>: 리스트 타입 모든 식 & value

Grammar Expression - IF

- if

```
if (<statement>) {...}
```

- else if

```
elif (<statement>) {...}
```

- else

```
else {...}
```

Grammar Expression - FOR

- range

```
for <variable> in range(<start>, <end>, <step>) {...}
```

- while

```
for (<statement>) {...}
```

- list

```
for <variable> in <list> {...}
```

Function

Token Description

<type>: 자료형

<name>: 사용되는 모든 문자열 토큰 (문자 + (0문자 이상의 문자열 & 0개 이상의 숫자))

<init_value>와 <value>는 value 가능

<varname>, <funcname>: <name>과 같음

<argv>: <list>와 같음

Grammar Expression

- function definition

```
func<<type>> <name>(<argv>) {...}
```

- function call

```
<funcname>(<argv>)
```

- struct function call

```
<varname>.<funcname>(<argv>)
```

Structure (>v0.2)

Token Description

<name>: 사용되는 모든 문자열 토큰 (문자 + (0문자 이상의 문자열 & 0개 이상의 숫자))

<varname>, <funcname>: <name>과 같음

Grammar Expression

아직 제대로 된 계획이 없습니다. v0.2까지는 구현 계획이 없습니다.

- struct definition

```
struct <varname> {...}
```

Miscellaneous

- Comment: only line-comment //

```
// 안녕하세요 (o)
```

```
out() // skdjfls (x)
```

- 모든 기능은 Builtin, 세미콜론은 없든 말든 상관 없음

연락처

- Wopslang Manager(Github): @RedoC-github, @pokmui, @codenstory
 - 추가 매니저는 합의를 통해 정해집니다.
- Slack <https://wopslangcommunity.slack.com/>
- <https://github.com/Wopslang> 는 깃허브 팀 링크입니다.