

기본 문법

변수

```
# 변수 정의
x = 10
y = "안녕하세요, Julia!"
z = 3.14
```

함수

```
# 함수 정의
function greet(name)
    return "안녕하세요, $name!"
end
```

```
# 함수 호출
println(greet("월드"))
```

제어 흐름

```
If-Else
a = 5
if a > 0
    println("양수입니다")
elseif a < 0
    println("음수입니다")
else
    println("0입니다")
end
```

```
For 루프
for i in 1:5
    println(i)
end
```

```
While 루프
i = 1
while i ≤ 5
    println(i)
    i += 1
end
```

데이터 구조

배열

```
# 1차원 배열
arr = [1, 2, 3, 4, 5]
```

```
# 다차원 배열
matrix = [1 2; 3 4]
```

딕셔너리

```
# 딕셔너리 정의
dict = Dict{"이름" => "Julia", "버전" => 1.9}
```

```
# 요소 접근
println(dict["이름"])
```

문자열

```
# 문자열 생성
str1 = "Hello"
str2 = "Julia"
```

```
# 문자열 연결
combined_str = str1 * " " * str2
println(combined_str)
```

```
# 문자열 보간
name = "Alice"
message = "안녕하세요, $(name)!"
println(message)
```

튜플

```
# 튜플 생성
tup = (1, "hello", 3.14)
println(tup[2]) # "hello"
```

```
# 튜플 언패킹
a, b, c = tup
println(a) # 1
```

구조체 (Structs)

```
# 구조체 정의
struct Point
    x::Int
    y::Int
end
```

```
# 구조체 인스턴스 생성
p = Point(10, 20)
println(p.x) # 10
```

고급 기능

에러 처리

```
# try-catch 블록
function divide(a, b)
    try
        return a / b
```

```
    catch e
        if isa(e, DivideError)
            println("0으로 나눌 수 없습니다!")
        else
            println("에러 발생: $e")
        end
    end
end
```

```
divide(10, 2)
divide(10, 0)
```

모듈 및 패키지

```
# 모듈 사용
using LinearAlgebra
println(det([1 2; 3 4]))
```

```
# 패키지 추가 (REPL에서)
# using Pkg
# Pkg.add("DataFrames")
```

브로드캐스팅

```
# 배열에 스칼라 값 더하기
arr = [1, 2, 3]
result = arr .+ 10
println(result) # [11, 12, 13]
```

컴프리헨션

```
# 배열 컴프리헨션
squares = [i^2 for i in 1:5]
println(squares) # [1, 4, 9, 16, 25]
```

```
# 딕셔너리 컴프리헨션
dict_comp = Dict{i => i^2 for i in 1:3}
println(dict_comp) # Dict{3 => 9, 2 => 4, 1 => 1}
```

유용한 명령어

```
# 함수에 대한 도움말 얻기
?println
```

```
# 패키지 관리자
using Pkg
Pkg.add("ExamplePackage")
Pkg.status()
```