Orus Simplified Syntax Cheatsheet (v0.2.0+)

```
Orus Simplified Syntax Cheatsheet (v0.2.0+)

Getting Started
---------------
fn main:
    print("Hello, Orus!")

Variables and Mutability
------------------------
let number = 5           # inferred as i32
let flag: bool = true    # explicitly bool
let mut count = 0        # mutable

Constants
---------
pub const LIMIT: i32 = 10
fn main:
    for i in 0..LIMIT:
        print(i)

Control Flow
------------
if n > 0:
    print("positive")
elif n == 0:
    print("zero")
else:
    print("negative")
print("ok") if x == 1 elif x == 2 else print("fallback")
let label = x > 0 ? "positive" : "non-positive"
for i in 0..5:
    print(i)

Functions
---------
fn add(a: i32, b: i32) -> i32:
    a + b

fn greet(name: string):
    print("Hello, {}!", name)

Structs and Enums
-----------------
struct Point:
    x: i32
    y: i32

enum Status:
    Ok
```

Orus Simplified Syntax Cheatsheet (v0.2.0+)

```
    NotFound
    Error(message: string)

Pattern Matching
----------------
match value:
    0: print("zero")
    1: print("one")
    _: print("other")

Error Handling
--------------
try:
    let x = 10 / 0
catch err:
    print("Error: {}", err)

Arrays and Generics
-------------------
let nums: [i32; 3] = [1, 2, 3]
let zeros = [0; 5]
let dynamic: [i32] = []
push(dynamic, 42)
pop(dynamic)

fn identity<T>(x: T) -> T:
    x

Modules
-------
use math:*
use math: sin, cos, tan
use datetime as dt

Built-ins
---------
print("x = {}", x)
push(arr, value)
pop(arr)
len(arr)
sorted(arr)
substring(s, start, len)
input(prompt)
type_of(x)
int("42")
float("3.14")
range(1, 5)
sum(arr)
min(arr)
max(arr)
```

Orus Simplified Syntax Cheatsheet (v0.2.0+)

```
timestamp()

Type System
-----------
let x = 10          # i32
let y = 3.14        # f64
let b: u32 = x as u32
let z: string = x as string

Operators precedence: ! > * > + > << > & > | > == > and > or > ? :

Common Mistakes:
- Use parentheses with not and ternary
- Chained comparisons invalid
```