Cheatography

Zig Builtins Cheat Sheet

by nyc (nyc) via cheatography.com/202869/cs/43118/

Legend

Γ comptime	σ []const u8	τ type
⊥ void	@V Vector	∀ anytype

when ∀ is returned it refers to the type of the ∀ arguments

Compiler

```
@src() Source Loc ation
@setAl ign Stack(Γ alignment: u29) ⊥
@setCold(Γ is_cold: bool) ⊥
@setEv alB ran chQ uota(Γ new_quota: u32) ⊥
@setRu nti meS afety(Γ safety_on: bool) ⊥
@inΓ() bool
@trap() noreturn
@break point() ⊥
@frame Add ress() usize
@hasDecl(Γ Contai ner:τ, Γ name: σ) bool
@hasFi eld(Γ Contai ner:τ, Γ name: σ) bool
@import(Γ path: σ)τ
@expor t(decl, Γ opts: Export Opt ions) ⊥
@exter n(T:τ, Γ opts: Extern Opt ions) T
@prefe tch (ptr: ∀, Γ opts: Prefet chO ptions) ⊥
@retur nAd dress() usize
@field (lhs: ∀, Γ name: σ) (field)
@field Par ent Ptr(Γ fname: σ, ptr: *T) ∀
@alignOf(Γ T:τ) Γ_int
@sizeOf(Γ T:τ) Γ_int
@bitSi zeOf(Γ T:τ) Γ_int
@bitOf fse tOf(Γ T:τ, Γ field: σ) Γ_int
@bitOf fse tOf(Γ T:τ, Γ field: σ) Γ_int
@call(mod: CallMo difier, func: ∀, args: ∀) ∀
@tagNa me( value: ∀) [:0]const u8
@union Init(Γ Union:τ, Γ active: σ, init) Union
@embed File(Γ path: σ) *const [N:0]u8
@compi leE rror(Γ msg: σ) noreturn
@error Nam e(err: anyerror) [:0]const u8
@error Ret urn Trace() ?*Stac kTrace
@panic (me ssage: σ) noreturn
@This() τ
```

Data and Logic

```
@memcp y(n oalias dst, noalias src) ⊥
@memse t(dest, elem) ⊥
@popCo unt (op erand: ∀) ∀
@shlEx act (value: T, shift_amt: Log2T) T
@shlWi thO ver flow(a: ∀, amt: Log2T) struct { ∀, u
1 }
@shrEx act (value: T, shift_amt: Log2T) T
@byteS wap (op erand: ∀) T
@bitRe ver se( int eger: ∀) T
@clz(o perand: ∀) ∀
@ctz(o perand: ∀) ∀
```

C Interop

```
@cDefine(Γ name: σ, value) ⊥
@cImpo rt( expr)τ
@cIncl ude(Γ path: σ) ⊥
@cUndef(Γ name: σ) ⊥
@cVaAr g(o perand: *VaList, Γ T:τ) T
@cVaCo py(src: *VaList) VaList
@cVaEn d(src: *VaList) ⊥
@cVaSt art() VaList
```

GPU

```
@workG rou pId(Γ dim: u32) u32
@workG rou pSize(Γ dim: u32) u32
@workI temId(Γ dim: u32) u32
```

Atomics

```
@fence (order: AO) ⊥
@atomicLoad
(Γ T:τ, p: *const T, Γ order: AO) T
@atomicStore
(Γ T:τ, p: *T, val: T, Γ order: AO) ⊥
@atomicRmw
(Γ T:τ, p: *T, Γ op: ARO, op: T, Γ order: AO) T
@cmpxchgWeak
(Γ T:τ, p: *T, exp: T, new: T, scs: AO, fail: AO) ?T
@cmpxchgStrong
(Γ T:τ, p: *T, exp: T, new: T, scs: AO, fail: AO) ?T
```

AO = std.builtin.AtomicOrder
RMO = std.builtin.AtomicRmwOp

Casting

```
@as(Γ T:τ, expr) T
@bitCa st( value: ∀) ∀
@ptrCa st( value: ∀) ∀
@ptrFr omI nt( add ress: usize) ∀
@addrS pac eCa st(ptr: ∀) ∀
@align Cas t(ptr: ∀) ∀
@enumF rom Int (in teger: ∀) ∀
@intCa st(int: ∀) ∀
@intFr omB ool (value: bool) u1
@intFr omE num (en um_ tag union: ∀) ∀
@intFr omE rro r(err: ∀) Int
@intFr omF loa t(f loat: ∀) ∀
@intFr omP tr( value: ∀) usize
@float Fro mIn t(int: ∀) ∀
@float Cas t(v alue: ∀) ∀
@error Fro mIn t(v alue: Int) anyerror
@volat ile Cas t(v alue: ∀) DestType
@const Cas t(v alue: ∀) DestType
@error Cas t(v alue: ∀) ∀
@trunc ate (in teger: ∀) ∀
```

SIMD

```
@Vecto r(len: Γ_int, Elemen t:τ)τ
@shuffle(Γ E:τ, a: @V, b: @V, Γ mask: @V) @V
@splat (sc alar: ∀) ∀
@reduce(Γ op: ReduceOp, value: ∀) E
@select(Γ T:τ, pred: @V, a: @V, b: @V) @V
```

Math

```
@setFl oat Mode(Γ mode: FloatMode) ⊥
@max(a: T, b: T) T
@min(a: T, b: T) T
@addWi thO ver flow(a: ∀, b: ∀) struct { ∀, u1 }
@mulWi thO ver flow(a: ∀, b: ∀) struct { ∀, u1 }
@subWi thO ver flow(a: ∀, b: ∀) struct { ∀, u1 }
@divEx act (num: T, denom: T) T
@divFl oor (num: T, denom: T) T
```

Math (cont)

```
@divTr unc (numr: T, denom: T) T
@mulAdd(Γ T:τ, a: T, b: T, c: T) T
@mulAdd(Γ T:τ, a: T, b: T, c: T) T
@abs(v alue: ∀) ∀
@mod(num: T, denom: T) T
@rem(num: T, denom: T) T
@sqrt( value: ∀) ∀
@sin(v alue: ∀) ∀
@cos(v alue: ∀) ∀
@tan(v alue: ∀) ∀
@exp(v alue: ∀) ∀
@exp2( value: ∀) ∀
@log(v alue: ∀) ∀
@log2( value: ∀) ∀
@log10 (value: ∀) ∀
@floor (value: ∀) ∀
@ceil( value: ∀) ∀
@trunc (value: ∀) ∀
@round (value: ∀) ∀
```

*WASM

```
@wasmM emo ryS ize(i: u32) u32
@wasmM emo ryG row(i: u32, d: u32) i32
```

Types

```
@Type(Γ info: Type)τ
@TypeO f(o perand)
@typeI nfo(Γ T:τ) Type
@typeN ame (T:τ) *const [N:0]u8
```