

WASM Module

```
graph TD; A[WASM Module] --> B[Text format]; A --> C[Binary format];
```

Sections

```
graph TD; A[WASM Module] --> B[Sections];
```

Known section
Custom sections

```
graph TD; A[Sections] --> B[Known section]; A --> C[Custom sections];
```

checked at instantiation for validation

```
graph TD; A[Known section] --> B[checked at instantiation for validation];
```

not checked at instantiation for validation

```
graph TD; A[Custom sections] --> B[not checked at instantiation for validation];
```

- Known section
- Type
- Import
- Function
- Table
- Memory

byte aligned memory

→ We start at this memory address

Preamble

```
00000000: 0061 736d      ; WASM_BINARY_MAGIC
00000004: 0100 0000      ; WASM_BINARY_VERSION
```

List of all function signatures

```
(type (;0;) (func))
(type (;1;) (func (param i32)))
(type (;2;) (func (param i32) (result i64)))
(type (;3;) (func (param i32 i32)))
(type (;4;) (func (param i32) (result i32)))
(type (;5;) (func (param i32 i32) (result i32)))
```

All imports for the module

```
(import "wasi_snapshot_preview1" "fd_write"
(func $_ZN4wasi13lib_generated22wasi_snapshot
_preview18fd_write17hd912d160882aaf3eE
(type 8)))
```

List of all the functions

```
(func $_start (type 0)
(local i32)
block ;; label = @1
call $_original_main
local.tee 0
i32.eqz
br_if 0 (;@1;)
local.get 0
call $exit
unreachable
end)
```

Typed array of references that can't be stored in linear memory

Linear memory for the module

```
(module
(memory (export "mem") 1)
(func (export "fill") (param i32 i32 i32)
local.get 0
local.get 1
local.get 2
memory.fill))
```

Start

Export

Global

Code

Element

Data

Function called when you run the module

list of all objects that will get returned to the host

definition of global variables

Body of each function

declares the data that gets loaded into the module's Table

data that gets loaded into the module's linear memory