

Public Method Types

Methods can be called externally by using the `@view({})` or `@call({})` decorators within the contract class which will expose the method in the compiled WASM bytecode to be called by any other NEAR Account. Whenever a method is declared in your contract class without these decorators and is called by another NEAR Account, a `MethodNotFound` error will be thrown.

tip If you need a contract to call itself, you can mark the function with these decorators but add the `@privateFunction: true` [annotation in the decorator parameters](#) so that it will panic if called from anything but the contract itself. A basic usage of this would look like the following:

```
@ NearBindgen ( { } ) export
```

```
class
```

```
MyContractStructure
```

```
{ @ call ( { } )
```

```
// or @view({}) some_method ( { parameter_a , parameter_b } )
```

```
{ // .. method logic here } }
```

snake_case vs camelCase We recommend using snake_case for method names, contrary to the camelCase convention in JavaScript. This is because the method names for a majority of contracts in the NEAR ecosystem use snake_case , and it is easier to use the same convention for all contracts. Where this would expose some_method from the WASM binary and allow it to be called externally. [Edit this page](#) Last updated on Jan 2, 2024 by Honza Smid Was this page helpful? Yes No

[Previous Collections](#) [Next Private Methods](#)