

Constants

Constants are values that are bound to a name and cannot change. They are always immutable. In Rust, constants are declared with the `const` keyword. Unlike variables declared with the `let` keyword, constants must be annotated with their type.

Constants are valid for the entire length of the transaction. They are essentially inlined wherever they are used, meaning that their value is copied directly into whatever context invokes them.

Since their value is hardcoded, they can save on gas cost as their value does not need to be fetched from storage.

Learn More

- [Rust docs - Constant items](#)
- [Solidity docs - Constant variables](#)

src/lib.rs

note This code has yet to be audited. Please use at your own risk. // Only run this as a WASM if the export-abi feature is not set.

```
#![cfg_attr(not(any(feature =
```

```
"export-abi", test)), no_main)] extern
```

```
crate
```

```
alloc ;
```

```
use
```

```
alloc :: vec ; use
```

```
alloc :: vec :: Vec ;
```

```
use
```

```
stylus_sdk :: alloy_primitives :: Address ; use
```

```
stylus_sdk :: prelude :: * ; use
```

```
stylus_sdk :: storage :: StorageAddress ;
```

```
const
```

```
OWNER :
```

```
& str
```

```
=
```

```
"0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045" ;
```

[storage]

[entrypoint]

```
pub
```

```
struct
```

```
Contract
```

```
{ owner :
```

```
StorageAddress , }
```

[public]

```
impl
Contract
{ pub
fn
init ( & mut
self )
->
Result < ( ) ,
Vec < u8
{ // Parse the const &str as a local Address variable let owner_address =
Address :: parse_checksummed ( OWNER ,
None ) . expect ( "Invalid address" ) ;
// Save the result as the owner self . owner . set ( owner_address ) ;
Ok ( ( ) ) } pub
fn
owner ( & self )
->
Result < Address ,
Vec < u8
{ let owner_address =
self . owner . get ( ) ;
Ok ( owner_address ) } }
```

Cargo.toml

```
[ package ] name
=
"stylus_constants_example" version
=
"0.1.7" edition
=
"2021" license
=
"MIT OR Apache-2.0" keywords
=
[ "arbitrum" ,
"ethereum" ,
```

```
"stylus" ,  
"alloy" ]  
[ dependencies ] alloy-primitives  
=  
"=0.7.6" alloy-sol-types  
=  
"=0.7.6" mini-alloc  
=  
"0.4.2" stylus-sdk  
=  
"0.6.0" hex  
=  
"0.4.3"  
[ dev-dependencies ] tokio  
=  
{  
version  
=  
"1.12.0" ,  
features  
=  
[ "full" ]  
} ethers  
=  
"2.0" eyre  
=  
"0.6.8"  
[ features ] export-abi  
=  
[ "stylus-sdk/export-abi" ]  
[ lib ] crate-type  
=  
[ "lib" ,  
"cdylib" ]  
[ profile.release ] codegen-units  
=  
1 strip
```

=

true lto

=

true panic

=

"abort" opt-level

=

"s" [Edit this page](#) [Previous Variables](#) [Next Function](#)