Cairo Naming Conventions

It is advised to follow the Cairo naming conventions since it'll help both code writers and readers to have a known standard.

snake_case naming:

```
// Modules (Contract names) use snake_case
mod sample_module {}

// Functions and methods's names use snake_case
fn a_method() {
    // The same naming convention applies to variables too.
    let local_variable = 1234;
}

// Struct fields use snake_case
struct MyStruct {
    field_a: felt252,
    field_b: u32,
}
```

PascalCase naming:

```
// Types use PascalCase. This includes enums, structs and traits and
implementations
type FeltAlias = felt252;

// Type parameters are types too, so they also use PascalCase.
// Although they are generally single letters, like the `T` used here.
trait SomeAction<T> {
    fn trigger_action(self: T);
}

// Enum variants use PascalCase
enum MyEnum {
    VariantA: felt252,
    VariantB: (),
}
```

UPPER_CASE naming:

```
// Constants are an exception as they use UPPER_CASE naming
const A_CONSTANT: felt252 = 1234;
```

Combined Example

```
impl MyStructImpl of SomeAction<MyStruct> {
    fn trigger_action(self: MyStruct) {
        self.field_a.print();
        self.field_b.print();
    }
}
```