



# Integer manipulations

## The Perl 6 integer test report

The integer is in perl the most important type so we need to do our best to test it thoroughly. First we declare a constant to be used later and an integer without a value.

```
constant ABC = 10;
my Int $i;
nok $i.defined, 'T0';
ok $i ~~ Int, 'T1';
todo 'T2', 1;
ok $i !~~ Int, 'T3';
ok $i ~~ Cool, 'T4';
```

- ✓ **T0:** Declaration of an integer without an assignment of a value, should result in an undefinedness.
- ✓ **T1:** But when defined or not, its type value can be compared.
- ✗ **T2:** Someday we might make this not fail when wormhole modules are installed
- ✗ **T3:** The opposite test should fail
- ✓ **T4:** Also inherited classes should be compared successfully.

## Operations

Next a few tests to see if operations can be applied to the integer

```
$i = ABC;
is $i, 10, 'T5';
todo 'T6', 2;
is $i, 11, 'T7';
like $i.Str, /'not 10, but text'/, 'T8';
```

- ✓ **T5:** Initialize Int with the constant seems to have done well
- ✗2 **T6:** Incredible tests which will be true in the not so distant future
- ✗ **T7:** Integer is 10 not 11
- ✗ **T8:** test some text

```
for ^10 {
  ok $^a < 7, 'T9';
}
```

- ✓7 ✗3 **T9:** test a series of numbers against 5