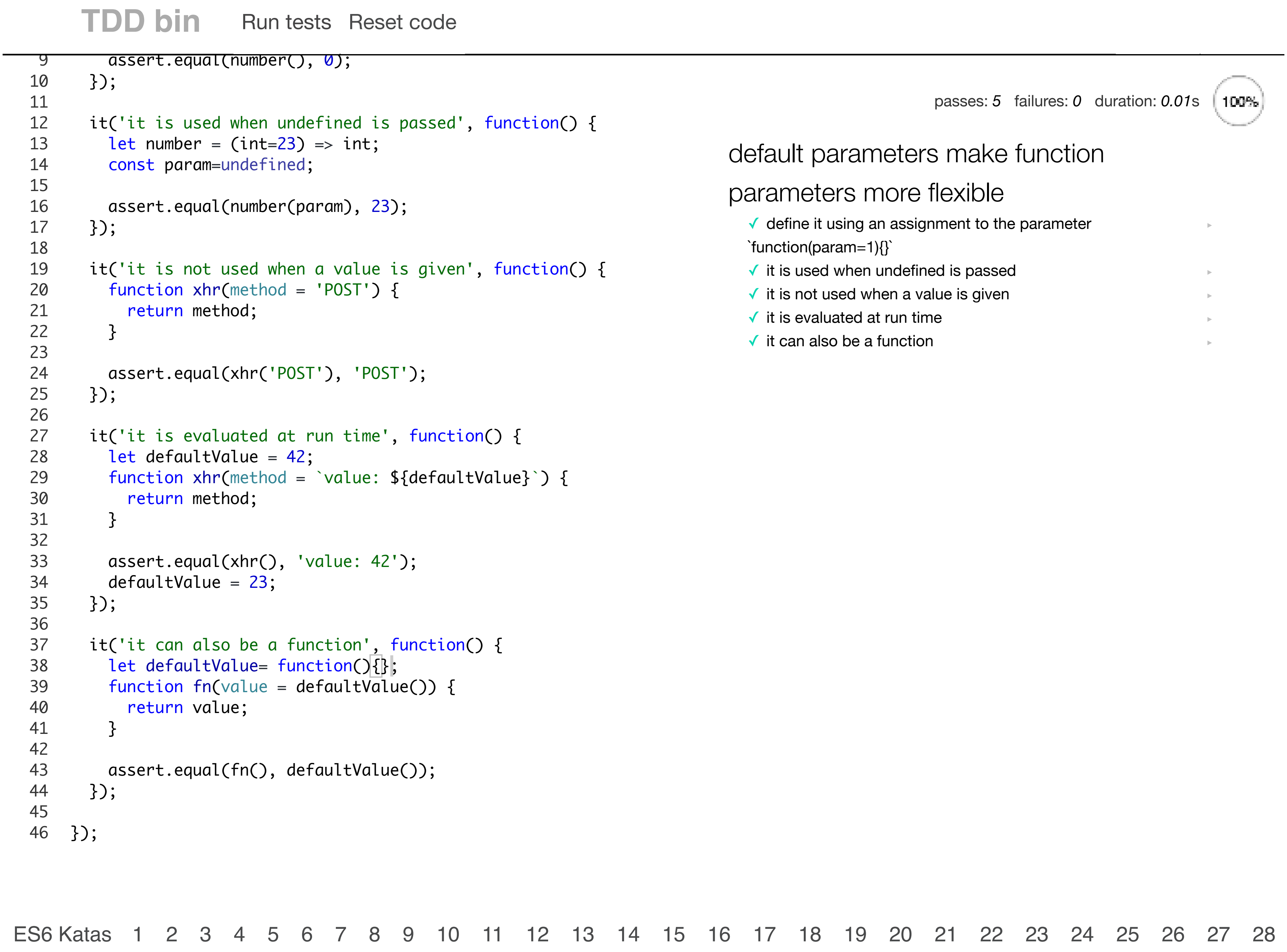

```
    assert.equal(number(), 0);
  });

  it('it is used when undefined is passed', function() {
    let number = (int=23) => int;
    const param=undefined;

    assert.equal(number(param), 23);
  });

  it('it is not used when a value is given', function() {
    function xhr(method = 'POST') {
      return method;
    }

    assert.equal(xhr('POST'), 'POST');
  });

  it('it is evaluated at run time', function() {
    let defaultValue = 42;
    function xhr(method = `value: ${defaultValue}`) {
      return method;
    }

    assert.equal(xhr(), 'value: 42');
    defaultValue = 23;
  });

  it('it can also be a function', function() {
    let defaultValue= function(){};
    function fn(value = defaultValue()) {
      return value;
    }

    assert.equal(fn(), defaultValue());
  });

});
```

passes: *5* failures: *0* duration: *0.01*s 100%

default parameters make function parameters more flexible

- ✓ define it using an assignment to the parameter `function(param=1){}`
- ✓ it is used when undefined is passed
- ✓ it is not used when a value is given
- ✓ it is evaluated at run time
- ✓ it can also be a function

ES6 Katas 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28