Values

What are Values?

```
int i = 2;

int main(int ac, const char *av[], const char *const *envp) {
    return ((const char *)main)[
        (long long signed int)main^ (unsigned) (
            ac * (long)(av) +
                (double) (char) (short) (long) (long long) envp
    )];
}
```

Values runtime creatures, passed as parameter, returned from functions, stored in variables, e.g., 2

Function values special kind of values, e.g,

Types Classification of values, mostly compile time creatures, e.g., `int`

Variables Where values are stored, mostly runtime, e.g. `i`

Dynamic Typing roughly, types are attached to values and managed at run time

Forms of Values

Atomic do not contain other values: `2`

Compound composed from other (compound or atomic) values, e.g., instance of a `struct` type, an array in Pascal, tuples in SML etc.

Atomic Values

Characterize the programming language

Numeric extremely typifying; may be non-scalar, e.g., primitive 2D points; lots of operations.

1. Only approximate the sets $\mathbb{Z}$ and $\mathbb{S}$;
2. Balance efficiency (hardware match) and portability

Boolean typifying, less essential than numeric, several operations

Symbolic: only operation is comparison. Related terms alphabet

- Any values in $\Sigma = \{a, b, c\}$
- Bit values, drwan from $\Sigma_2 = \{0, 1\}$
- The `ok` value (also written `()`); the only value of the unit type.