

## Primitive Types

$\langle singleQuote \rangle ::= ' '$

$\langle doubleQuote \rangle ::= ''$

$\langle int \rangle ::= [ \text{integer} ]$

$\langle bool \rangle ::= \text{'true'} \mid \text{'false'}$

$\langle char \rangle ::= \langle singleQuote \rangle [ \text{character} ] \langle singleQuote \rangle$

$\langle string \rangle ::= \langle doubleQuote \rangle [ \text{character}^* ] \langle doubleQuote \rangle$

$\langle null \rangle ::= \text{'null'}$

## Algebraic Data Types and Type-Traits

$\langle adt \rangle ::= \text{'type'} \langle ident \rangle \{ '[ \langle ident \rangle ':' \langle type \rangle [ ',' \langle ident \rangle ':' \langle type \rangle ]^* ] ' \}$

$\langle trait \rangle ::= \text{'trait'} \{ '[ \langle type \rangle [ ',' \langle type \rangle ]^* ] ' \}$

## Types

$\langle type \rangle ::= \text{'int'} \mid \text{'bool'} \mid \text{'char'} \mid \text{'string'} \mid \text{'null'}$   
|  $\langle type \rangle \text{'->'} \langle type \rangle$   
|  $\text{'('} [ \langle type \rangle [ ',' \langle type \rangle ]^* \text{' )' } \text{'->'} \text{'('} [ \langle type \rangle [ ',' \langle type \rangle ]^* \text{' )' }$   
|  $\text{'List'} \mid \text{'Array'} \text{'['} \langle type \rangle \text{']'}$   
|  $\text{'Tuple'} \text{'['} \langle type \rangle [ ',' \langle type \rangle ]^* \text{']'}$   
|  $\text{'Dict'} \text{'['} \langle type \rangle \text{' , ' } \langle type \rangle \text{']'}$   
|  $\langle ident \rangle$

## Arithmetic and Boolean Operators

$\langle arithOp \rangle ::= \text{'+'} \mid \text{'-'} \mid \text{'*'} \mid \text{'/'} \mid \text{'%'}$

$\langle boolOp \rangle ::= \text{'<'} \mid \text{'>'} \mid \text{'<='} \mid \text{'>='} \mid \text{'!'}$  |  $\text{'!='} \mid \text{'=='}$  |  $\text{'\&\&'} \mid \text{'||'}$

$\langle op \rangle ::= \langle arithOp \rangle \mid \langle boolOp \rangle$

## Functions

$\langle arg \rangle ::= \langle ident \rangle ':' \langle type \rangle [ \text{'='} \langle atom \rangle ]$

$\langle templateTypes \rangle ::= \text{'['} \langle type \rangle ':' \langle ident \rangle [ ',' \langle type \rangle ':' \langle ident \rangle ]^* \text{']'}$

$\langle prog \rangle ::= [ \text{'fn'} \langle ident \rangle [ \langle templateTypes \rangle ] \text{'('} [ \langle arg \rangle [ ',' \langle arg \rangle ]^* \text{' )' } \text{'->'} \langle type \rangle ] \text{'='} \langle smp \rangle \text{' ; ' } ]^*$

## Pattern Matching and Switches

$\langle match \rangle ::= \text{'match' } \langle ' \rangle \langle ident \rangle \langle ' \rangle \{ \text{'case' } \langle type \rangle \text{'=>' } \langle smp \rangle [ \text{' , ' } \text{'case' } \langle type \rangle \text{'=>' } \langle smp \rangle ]^* \}$

$\langle switch \rangle ::= \text{'switch' } \langle ' \rangle \langle atom \rangle \langle ' \rangle \{ \text{'case' } \langle atom \rangle \text{'=>' } \langle smp \rangle [ \text{' , ' } \text{'case' } \langle atom \rangle \text{'=>' } \langle smp \rangle ]^* \}$

$\langle matchSwitch \rangle ::= \langle match \rangle \mid \langle switch \rangle$

## Expressions

$\langle smp \rangle ::= \langle utight \rangle [ \langle op \rangle \langle utight \rangle ]$   
|  $\text{'if' } \langle ' \rangle \langle smp \rangle \langle ' \rangle [ \text{'else' } \langle smp \rangle ]$   
|  $\text{'List' } \mid \text{'Tuple' } \mid \text{'Array' } \{ \langle ' \rangle [ \langle smp \rangle [ \text{' , ' } \langle smp \rangle ]^* ] \langle ' \rangle \}$   
|  $\text{'Dict' } \{ \langle ' \rangle [ \langle smp \rangle \text{' : ' } \langle smp \rangle [ \text{' , ' } \langle smp \rangle \text{' : ' } \langle smp \rangle ]^* ] \langle ' \rangle \}$   
|  $\langle matchSwitch \rangle$   
|  $\langle trait \rangle$   
|  $\langle adt \rangle$   
|  $\langle prog \rangle$   
|  $\langle arg \rangle \text{'=>' } \langle smp \rangle$

$\langle exp \rangle ::= \langle smp \rangle [ \text{' ; ' } \langle exp \rangle ]$   
|  $\text{'val' } \langle ident \rangle [ \text{' : ' } \langle type \rangle ] \text{' = ' } \langle smp \rangle \text{' ; ' } \langle exp \rangle$