

Haskell Expressions

Scott N. Walck

January 22, 2024

Every Haskell expression has a value and a type.

Expression	Value	Type
<code>4.5 + 4</code>	<code>8.5</code>	<code>Double</code>
<code>4.2</code>	<code>4.2</code>	<code>Double</code>
<code>False</code>	<code>False</code>	<code>Bool</code>
<code>True</code>	<code>True</code>	<code>Bool</code>
<code>"hello"</code>	<code>"hello"</code>	<code>[Char]</code>
<code>not True</code>	<code>False</code>	<code>Bool</code>
<code>length "hello"</code>	<code>5</code>	<code>Int</code>
<code>map</code>	can't show	<code>(a -> b) -> [a] -> [b]</code>
<code>not</code>	can't show	<code>Bool -> Bool</code>
<code>map not</code>	can't show	<code>[Bool] -> [Bool]</code>
<code>id</code>	can't show	<code>a -> a</code>

Compound expressions are expressions too.

Expression	Simple or compound?
<code>4.5 + 4</code>	compound expression
<code>4.2</code>	simple expression
<code>False</code>	simple expression
<code>True</code>	simple expression
<code>"hello"</code>	simple expression
<code>not True</code>	compound expression
<code>length "hello"</code>	compound expression
<code>map</code>	simple expression
<code>not</code>	simple expression
<code>map not</code>	compound expression
<code>id</code>	simple expression