

real_kinetic.com/|@real_kinetic



REALKINETIC

YOUR POTENTIAL REALIZED

```
-- | Equivalent to Purescript's `show`.
toString :: ∀ a. (Show a) => a -> String
toString = show
```

```
-- | Equivalent to Purescript's `<<<`.
compose :: ∀ a b c. (b -> c) -> (a -> b) -> (a -> c)
compose = (<<<)
```

```
-- | Equivalent to Purescript's `$`.
applyFn :: ∀ a b. (a -> b) -> a -> b
applyFn = ($)
```

```
-- | The Purescript equivalent is `id`.
identity :: ∀ a. a -> a
identity = id
```

```
-- | The Purescript equivalent is `const`.
always :: ∀ a b. a -> b -> a
always = const
```

```
-- | The Purescript equivalent is `Void`.
type Never = Void
```

And if you have or you're
already wanting more then
here a few resources.

```
-- | Equivalent to Purescript's `show`.  
toString :: ∀ a. (Show a) => a -> String  
toString = show
```

```
-- | Equivalent to Purescript's `<<<`.  
compose :: ∀ a b c. (b -> c) -> (a -> b) -> (a -> c)  
compose = (<<<)
```

```
-- | Equivalent to Purescript's `$`.  
applyFn :: ∀ a b. (a -> b) -> a -> b  
applyFn = ($)
```

```
-- | The Purescript equivalent is `id`.  
identity :: ∀ a. a -> a  
identity = id
```

```
-- | The Purescript equivalent is `const`.  
always :: ∀ a b. a -> b -> a  
always = const
```

```
-- | The Purescript equivalent is `Void`.  
type Never = Void
```