

Latte

- Code block: `true`
- Code syntax: `true`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Latte

- Code block: `true`
- Code syntax: `false`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Latte

- Code block: `false`
- Code syntax: `true`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Latte

- Code block: `false`
- Code syntax: `false`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Frappé

- Code block: true
- Code syntax: true

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Frappé

- Code block: `true`
- Code syntax: `false`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Frappé

- Code block: `false`
- Code syntax: `true`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Frappé

- Code block: `false`
- Code syntax: `false`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```


Macchiato

- Code block: true
- Code syntax: true

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Macchiato

- Code block: `true`
- Code syntax: `false`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Macchiato

- Code block: `false`
- Code syntax: `true`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Macchiato

- Code block: `false`
- Code syntax: `false`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Mocha

- Code block: true
- Code syntax: true

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Mocha

- Code block: true
- Code syntax: false

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Mocha

- Code block: `false`
- Code syntax: `true`

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```

Mocha

- Code block: false
- Code syntax: false

```
#import "/src/lib.typ": catppuccin, flavors, get-flavor

#let perms = ()
#for flavor in flavors.values() {
  for code-block in (true, false) {
    for syntax in (true, false) {
      perms.push((flavor: flavor, code-block: code-block, syntax: syntax))
    }
  }
}

#for p in perms [
  #show: catppuccin.with(p.flavor, code-block: p.code-block, code-syntax: p.syntax)
  = #get-flavor(p.flavor).name
  == Code block: #p.code-block
  == Code syntax: #p.syntax
]
```