```julia
abstract type Distribution end
```

$$\mathcal{P} \triangleq \{p(\cdot;\theta) : \theta \in \Theta\}$$

Distribution space

Define set $\mathcal{P}$ as the collection of all probability distributions. Emphasizing generality, this set definition describes all of the possible models to describe the behavior of phenomena (in our case, random variables). To quantify our understanding of the nature of events, we turn to the probability distribution.

$$\begin{gathered}\mathcal{P} = \mathcal{P}_M \cup \mathcal{P}_J \cup \mathcal{P}_C, \text{ where} \\ \mathcal{P}_M \triangleq \{p(\cdot;\boldsymbol{\theta}(X)); \boldsymbol{\theta}(X) \in \boldsymbol{\Theta}(X)\}, \\ \\ \mathcal{P}_J \triangleq \{p(\cdot;\boldsymbol{\theta}(X_1^n); \boldsymbol{\theta}(X_1^n) \in \boldsymbol{\Theta}(X_1^n)\}, \\ \\ \mathcal{P}_C \triangleq \{p(\cdot;\boldsymbol{\theta}(X|Y)); \boldsymbol{\theta}(X|Y) \in \boldsymbol{\Theta}(X|Y)\}\end{gathered}$$

Distribution subspaces

Specifically, the distribution space is composed of three categories of probability distributions:

1. **Marginal distribution** — describes a single random variable X
2. **Joint distribution** — describes a collection of random variables $X_1, \ldots, X_n$
3. **Partial** *or* **Conditional distribution** — describes a single random variable X given the random variable Y

To begin, we emphasize that the collection of random variables $X_1, \ldots, X_n$ can be conceptualized instead as a single random variable $X_1^n \triangleq (X_1, \ldots, X_n)$. Our development of probability distributions begins with a univariate focus, and readily extends into a multivariate analysis.

```julia
mutable struct Marginal <: Distribution

    p::AbstractDict

    Marginal(p::AbstractDict) = new(p)

    Marginal() = new(Dict())

```

```julia
end
```

$$p_X(x;\boldsymbol{\theta}(X)) = \begin{cases} \theta_1(X), & x = \mathsf{a}_1, \\ \theta_2(X), & x = \mathsf{a}_2, \\ & \vdots \\ \theta_m(X), & x = \mathsf{a}_m \end{cases}$$

Marginal distributions

```
p = Marginal distribution:
     θ₁, x = a₁
     θ₂, x = a₂
     ...
     θₘ, x = aₘ
```

```julia
p = Marginal(Dict(
        :a₁ => :θ₁,
        :a₂ => :θ₂,
        :a₃ => :θ₃,
        :aₘ => :θₘ)
    )
```

```julia
mutable struct Joint <: Distribution

    p::AbstractDict

    Joint(p::AbstractDict) = new(p)

    Joint() = new(Dict())

end
```

$$p(x_1^n;\boldsymbol{\theta}(X_1^n)) = \begin{cases} \theta_1(X_1^n), & (x_1,\ldots,x_n) = (\mathsf{a}_1^{(1)},\ldots,\mathsf{a}_1^{(n)}) \\ \theta_2(X_1^n), & (x_1,\ldots,x_n) = (\mathsf{a}_2^{(1)},\ldots,\mathsf{a}_1^{(n)}) \\ & \vdots \\ \theta_{m_1\cdots m_n}(X_1^n), & (x_1,\ldots,x_n) = (\mathsf{a}_{m_1}^{(1)},\ldots,\mathsf{a}_{m_n}^{(n)}) \end{cases}$$

Joint distributions

```
q = Joint distribution:
     θ₁, (x₁, ..., xₙ) = (a₁⁽¹⁾, ..., a₁⁽ⁿ⁾)
     θ₂, (x₁, ..., xₙ) = (a₂⁽¹⁾, ..., a₁⁽ⁿ⁾)
     ...
     θₘ, (x₁, ..., xₙ) = (aₘ⁽¹⁾, ..., aₘ⁽ⁿ⁾)
```

```julia
q = Joint(
        Dict((:a₁⁽¹⁾, :a₁⁽²⁾, :a₁⁽³⁾, :a₁⁽ⁿ⁾) => :θ₁,
             (:a₂⁽¹⁾, :a₁⁽²⁾, :a₁⁽³⁾, :a₁⁽ⁿ⁾) => :θ₂,
             (:a₃⁽¹⁾, :a₁⁽²⁾, :a₁⁽³⁾, :a₁⁽ⁿ⁾) => :θ₃,
```

```
        # ...
            (:aₘ⁽¹⁾, :aₘ⁽²⁾, :aₘ⁽³⁾, :aₘ⁽ⁿ⁾) => :θₘ)
    )
```

```
mutable struct Partial <: Distribution

    p::AbstractDict

    Partial(p::AbstractDict) = new(p)

    Partial() = new(Dict())

end
```

```
mutable struct Conditional <: Distribution

    p::AbstractDict

    Conditional(p::AbstractDict) = new(p)

    Conditional() = new(Dict())

end
```

$$p(x|y;\boldsymbol{\theta}(X|Y)) = \begin{cases} \theta_1(X|Y), & x = a_1|y = b_1 \\ \theta_2(X|Y), & x = a_2|y = b_1 \\ & \vdots \\ \theta_{nm}(X|Y), & x = a_n|y = b_m \end{cases}$$

Partial, Conditional distributions

```
r = Partial distribution:
     θ₁, x = a₁ | y = b₁
     θ₂, x = a₂ | y = b₁
     ...
     θₙₘ, x = aₙ | y = bₘ
```

```
r = Partial(
        Dict(((:a₁,), (:b₁,)) => :θ₁,
             ((:a₂,), (:b₁,)) => :θ₂,
             ((:a₃,), (:b₁,)) => :θ₃,
        # ...
             ((:aₙ,), (:bₘ,)) => :θₙₘ)
    )
```