

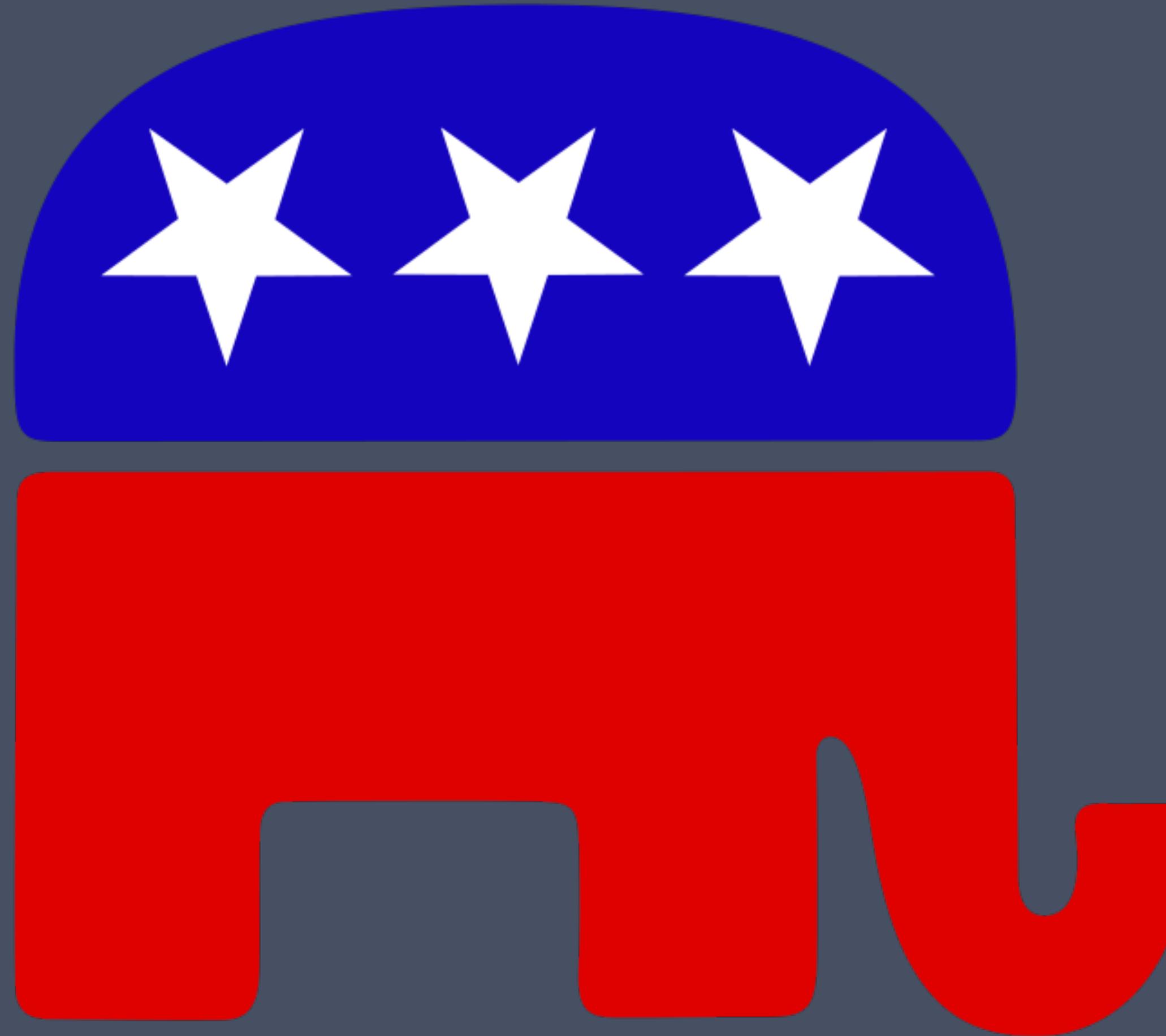
ROLLING RANDOM ROMANS





TAKE A JOURNEY

REPUBLICAN ROME





YOU

A classical painting depicting the Roman goddess Juno. She is seated on a chariot drawn by two peacocks. Juno is dressed in a white robe and holds a golden scepter. A young girl stands behind her, reaching up towards Juno's head. The background features a landscape with a rainbow and a figure on horseback.

JUNO

MARVASF

YOUR MISSION

ROMAN NAMING SYSTEM



MEN



PRAENOMEN

PUBLIUS CORNELIUS SCIPIO AFRICANUS

A painting depicting a scene from ancient Roman life, possibly a sacrifice or a procession, with figures in traditional attire.

NOMEN

PUBLIUS CORNELIUS SCIPIO AFRICANUS



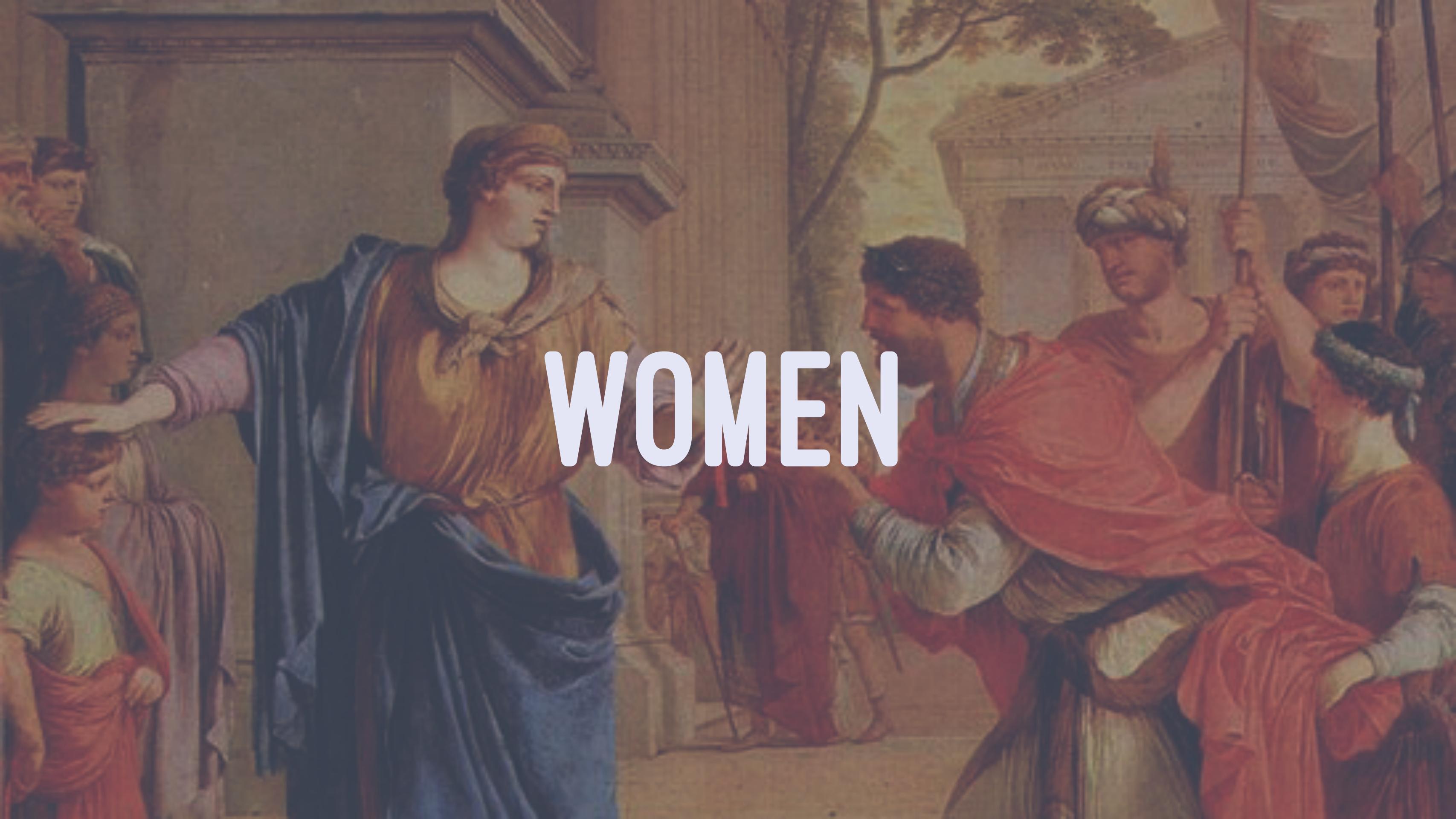
COGNOMEN

PUBLIUS CORNELIUS SCIPIO AFRICANUS



AGNOMEN

PUBLIUS CORNELIUS SCIPIO AFRICANUS



WOMEN



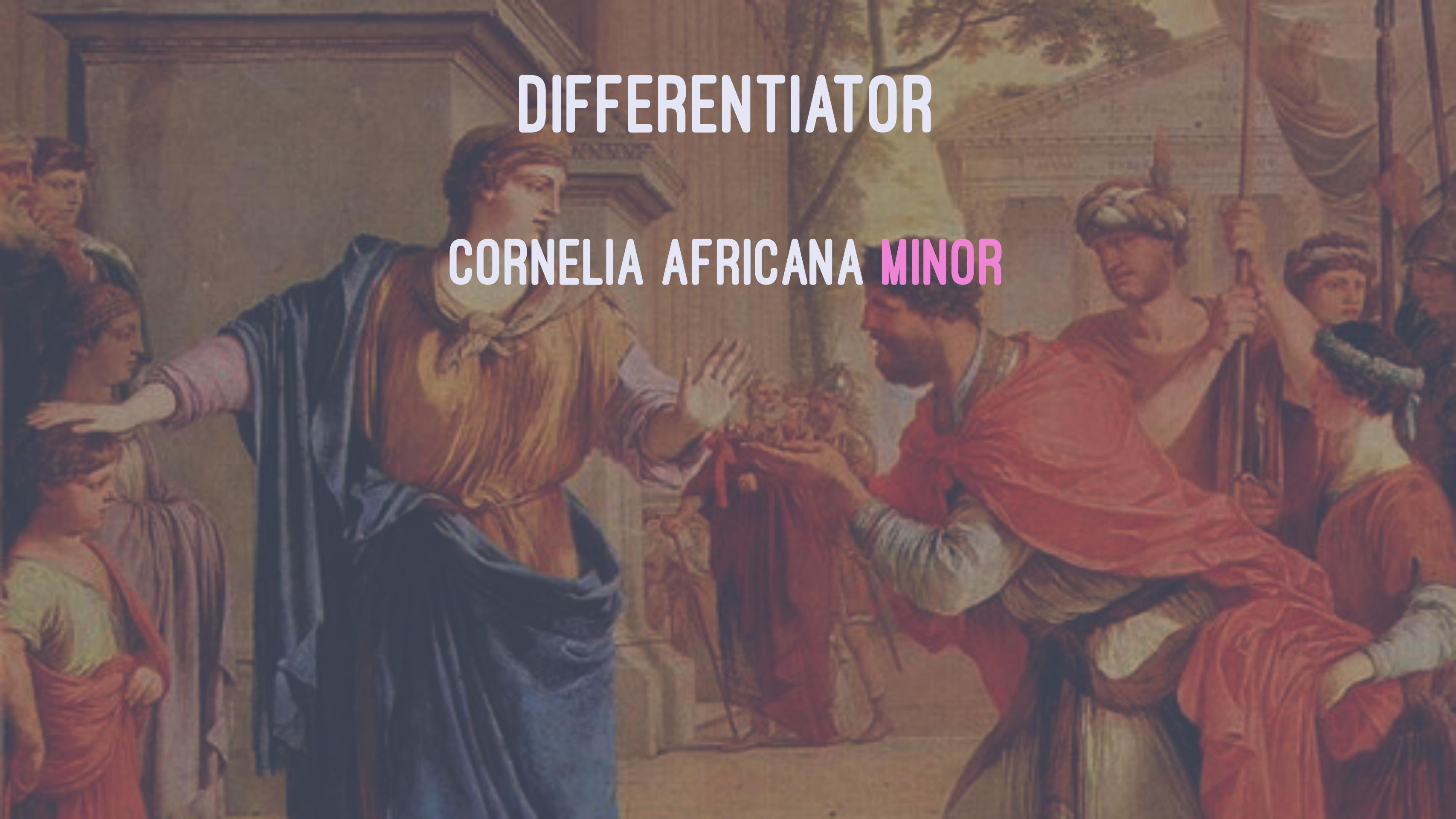
NOMEN

CORNELIA AFRICANA MINOR



COGNOMEN

CORNELIA AFRICANA MINOR



DIFFERENTIATOR

CORNELIA AFRICANA MINOR

OUR PROGRAM

MODEL

```
type alias Id = Integer
```

```
type alias Model =  
{  romans : Dict Id Roman  
,  nextId : Id  
}
```

ROMAN

```
type alias Roman =  
{ id : Id  
, clan : Clan  
, children : Children  
, name : Name  
}  
  
type Children = Children (List Roman)
```

NAME

```
type Name
= FemaleName (Maybe String) (Maybe String)
| MaleName String (Maybe String) (Maybe String)
```

```
type alias Clan =
{ name : String
, color : String
}
```

VIEW

```
viewRoman : Roman -> Html Msg
viewRoman roman =
    li [ style [ ( "color", roman.clan.color ) ] ]
        [ text (formattedName roman)
        , button [ onClick (GenerateChildFor roman) ] [ text "Bless with child" ]
        , viewChildren roman.children
        ]
```

UPDATE

```
type Msg
= GenerateChildFor Roman
| Birth Roman Roman

update : Msg -> Model -> ( Model, Cmd Msg )
update msg model =
  case msg of
    GenerateChildFor father ->
      ( model, Random.generate (Birth father) Random.Roman.roman )

    Birth father child ->
      -- append child to father
      ( { model | romans = updatedRomans }, Cmd.none )
```

**RANDOMNESS IN
A FUNCTIONAL
WORLD**

PURE FUNCTIONS

- > $2 + 2 = 4$
- > `floor 5.6 = 5`
- > `toUpper "abc" = "ABC"`

RANDOM IS INHERENTLY NOT PURE

```
rand(1000) # => 964
rand(1000) # => 592
rand(1000) # => 482
rand(1000) # => 872
rand(1000) # => 815
```

RANDOM AS A PURE FUNCTION

```
def seeded_rand(seed, max)
    srand(seed)
    rand(max)
end
```

```
seeded_rand(123, 1000) # => 510
```

PSEUDO-RANDOM

RANDOM IN ELM

EXPLICIT SEED

```
randomBool : (Bool, Random.Seed)
randomBool =
  Random.step (Random.initialSeed 123) (Random.bool)
```

COMMANDS

```
type Msg
= RequestRandom
| Generated Bool

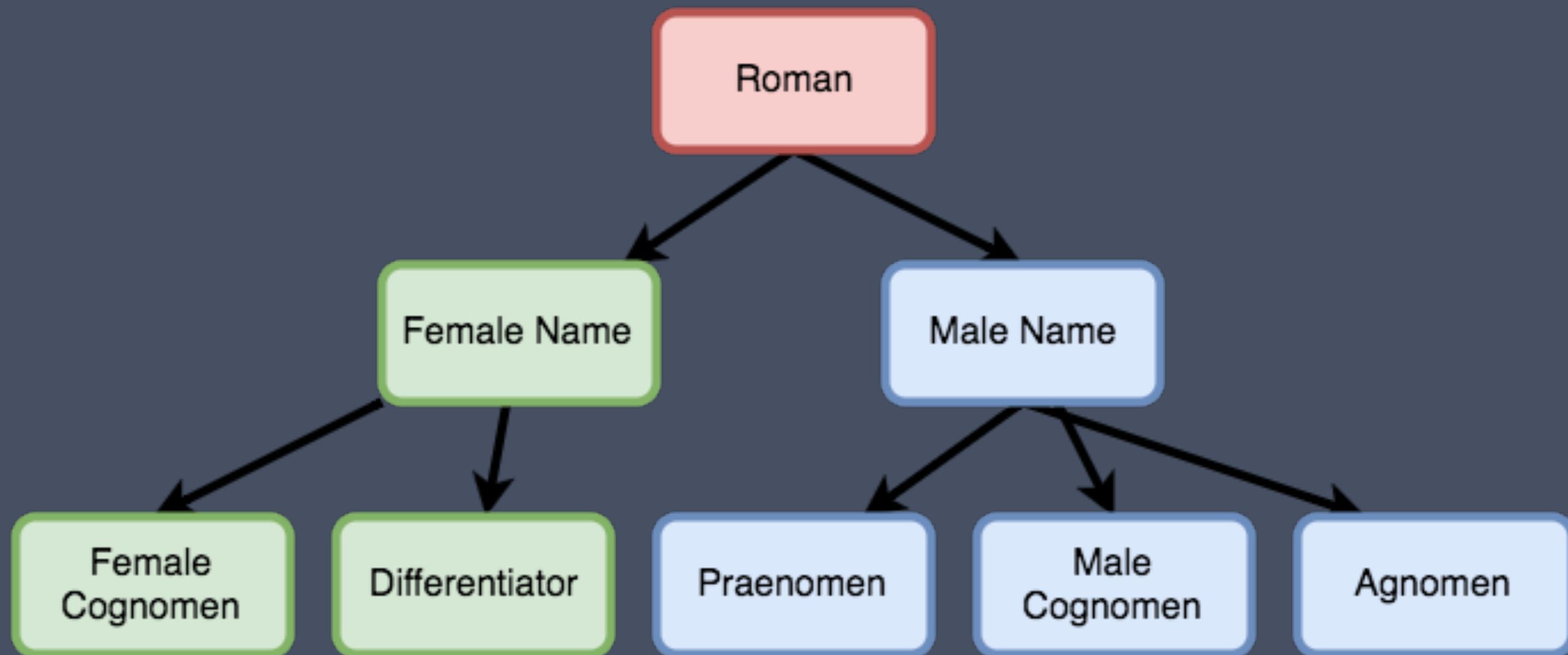
update : Msg -> Model -> (Model, Cmd Msg)
update msg model =
  case msg of
    RequestRandom -> (model, Random.generate Generated Random.bool)
    Generated bool -> ({ model | flag = bool }, Cmd.none)
```

COMPLEX VALUES

```
roman : Random.Generator Roman
roman =
  -- generator for romans with random
  -- * praenomen
  -- * cognomen
  -- * agnomen
  -- * differentiator
  -- * gender
```

BUILDING GENERATORS

WORK FROM THE BOTTOM UP



USING EXISTING GENERATOR

```
import Random.Extra

differentiator : Random.Generator (Maybe String)
differentiator =
let differentiators =
[ "Maior"
, "Minor"
, "Prima"
, "Tertia"
]
in
Random.Extra.sample differentiators
```

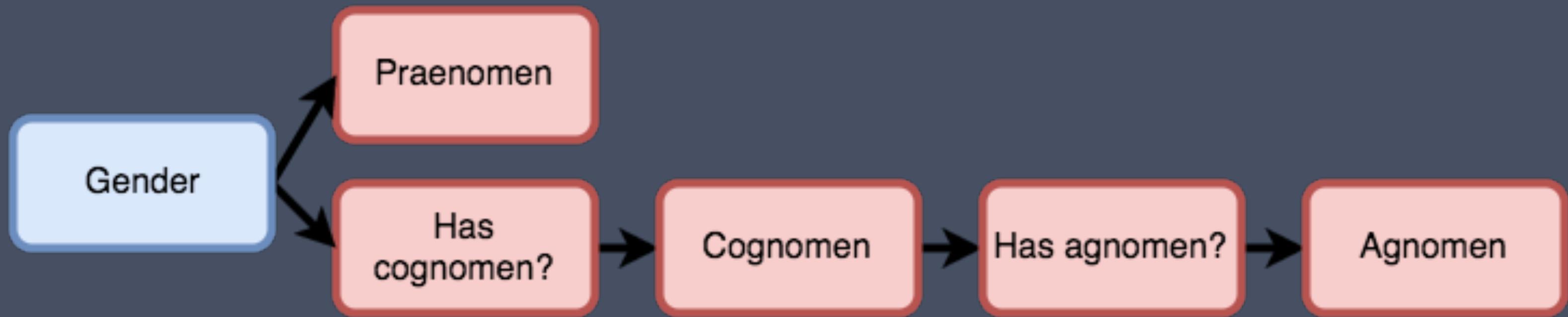
TRANSFORMING ROLLED VALUES

```
femaleName : Random.Generator Name  
femaleName =  
  Random.map (FemaleName Nothing) differentiator
```

COMBINING INDEPENDENT ROLLS

```
femaleName : Roman -> Random.Generator Name  
femaleName father =  
  Random.map2 FemaleName (femaleCognomen father) differentiator
```

CHAIN OF DEPENDENCIES



MALE NAME

```
maleName : Random.Generator Name  
maleName =  
  Random.map3 MaleName praenomen cognomen agnomen
```

AGNOMEN

```
agnomen : Maybe String -> Generator (Maybe String)
agnomen cog =
  case cog of
    Just _ ->
      Random.extra.sample ["Felix", "Cuncunctor", "Africanus"]
    Nothing ->
      Random.Extra.constant Nothing
```

CHAINING DEPENDENT ROLLS

```
import Random exposing(andThen)

maleName : Random.Generator Name
maleName =
    Random.map3 MaleName praenomen cognomen (cognomen `andThen` agnomen)
```

FIXING A BUG

```
import Random exposing(andThen)

maleNameFromCognomen : Maybe String -> Generator Name
maleNameFromCognomen cog =
    Random.map3 MaleName praenomen (Random.Extra.constant cog) (agnomen cog)

maleName : Generator Name
maleName =
    cognomen `andThen` maleNameFromCognomen
```

RANDOM PLAYBOOK

- > EXISTING GENERATORS
- > TRANSFORM GENERATORS WITH `Random.map`
- > COMBINE INDEPENDENT GENERATORS WITH `Random.map2` AND FRIENDS
- > COMBINE DEPENDENT GENERATORS WITH `Random.andThen`

Random.Extra

elm-community/random-extra

REMAINING STEPS

- > RINSE
- > REPEAT
- > PROFIT \$\$\$

EXTRA CREDIT

- > SOME PRAENOMINA ARE FAVORED BY A FAMILY
 - > COGNOMINA CAN BE HEREDITARY
 - > WEIGHT ROLLS WITH PERCENTAGES

DEMO TIME

[HTTPS://JOELQ.GITHUB.IO/ELM-CONF-DEMO](https://joelq.github.io/elm-conf-demo)

PRINCIPLES

- > TAKE ADVANTAGE OF COMMANDS
- > MAKE AS FEW CALLS TO Random.generate AS POSSIBLE
- > BUILD UP COMPLEX GENERATORS USING map, map2, andThen ETC
- > BUILD FROM THE BOTTOM UP

ABOUT ME

- > TWITTER: @JOELQUEN
- > GITHUB: @JOELQ
- > SLIDES: JOELQ/ELM-CONF-TALK
- > DEMO: JOELQ/ELM-CONF-DEMO

IMAGES CREDITS

- > DICE MICHAEL AND IS DISTRIBUTED UNDER THE CREATIVE COMMONS ATTRIBUTION 4.0 LICENSE.
- > SPQR SEE PAGE FOR AUTHOR [CC BY 3.0 ([HTTP://
CREATIVECOMMONS.ORG/LICENSES/BY/3.0](http://creativecommons.org/licenses/by/3.0)) OR PUBLIC DOMAIN]. VIA WIKIMEDIA COMMONS
- > REPUBLICAN ELEPHANT BY REPUBLICAN PARTY (UNITED STATES)

IMAGE CREDITS (CONTINUED)

- > ELEPHANTS AT ZAMA HENRI-PAUL MOTTE [PUBLIC DOMAIN OR PUBLIC DOMAIN]. VIA WIKIMEDIA COMMONS
- > SCIPIO AFRICANUS GIOVANNI BATTISTA TIEPOLO [PUBLIC DOMAIN OR PUBLIC DOMAIN]. VIA WIKIMEDIA COMMONS
- > CORNELIA AFRICANA LAURENT DE LA HYRE [PUBLIC DOMAIN OR