δίδωμι

Here's our value for `repo`:

```
repo = pwd() |> dirname |> dirname |> dirname |> dirname
repo
```

Present system

Present tense

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("present"), gmpVoice("active"),
διδωμι, kds))
```

Middle and passive voices (identical forms):

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("present"), gmpVoice("passive"),
διδωμι, kds))
```

Imperative

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
```

```
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")

tbl = md_imperativeconjugation(gmpTense("present"), gmpVoice("active"),
διδωμι, kds)
Markdown.parse(tbl)
```

Middle and passive voices (identical forms):

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")

tbl = md_imperativeconjugation(gmpTense("present"), gmpVoice("passive"),
διδωμι, kds)
Markdown.parse(tbl)
```

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")
vadj = GMFVerbalAdjective(
    gmpGender("neuter"), gmpCase("nominative"), gmpNumber(1)
)
vadjforms = generate(διδωμι, formurn(vadj), kds)

inf_act = GMFInfinitive(
    gmpTense("present"), gmpVoice("active")
)
inf_actforms = generate(διδωμι, formurn(inf_act), kds)

inf_pass = GMFInfinitive(
    gmpTense("present"), gmpVoice("passive")
)
inf_passforms = generate(διδωμι, formurn(inf_pass), kds)

actptcpl = participleslashline(διδωμι, gmpTense("present"),
gmpVoice("active"), kds)

mpptcpl = participleslashline(διδωμι, gmpTense("present"),
gmpVoice("middle"), kds)
```

```
mdlines = [
    "- **active infinitive**: $(inf_actforms[1])",
    "- **middle & passive infinitive**: $(inf_passforms[1])",
    "- **active participle**: $(actptcpl)",
    "- **middle & passive participle**: $(mpptcpl)",
    "- **verbal adjective**: $(vadjforms[1])"]
Markdown.parse(join(mdlines,"\n"))
```

Imperfect tense

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("imperfect"), gmpVoice("active"),
διδωμι, kds))
```

Middle and passive voices (identical forms):

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("imperfect"), gmpVoice("passive"),
διδωμι, kds))
```

Future tense

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")
```

```
Markdown.parse(md_conjugation(gmpTense("future"), gmpVoice("active"),
διδωμι, kds))
```

Middle voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("future"),
gmpVoice("middle"),διδωμι, kds))
```

Passive voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("future"),
gmpVoice("passive"),διδωμι, kds))
```

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")



inf_act = GMFInfinitive(
    gmpTense("future"), gmpVoice("active")
)
inf_actforms = generate(διδωμι, formurn(inf_act), kds)


inf_mdl = GMFInfinitive(
    gmpTense("future"), gmpVoice("middle")
)
inf_mdlforms = generate(διδωμι, formurn(inf_mdl), kds)


inf_pass = GMFInfinitive(
    gmpTense("future"), gmpVoice("passive")
)
inf_passforms = generate(διδωμι, formurn(inf_pass), kds)


actptcpl = participleslashline(διδωμι, gmpTense("future"),
gmpVoice("active"), kds)

midptcpl = participleslashline(διδωμι, gmpTense("future"),
gmpVoice("middle"), kds)

passptcpl = participleslashline(διδωμι, gmpTense("future"),
gmpVoice("passive"), kds)

mdlines = [
    "- **active infinitive**: $(inf_actforms[1])",
    "- **middle infinitive**: $(inf_mdlforms[1])",
    "- **passive infinitive**: $(inf_passforms[1])",
    "- **active participle**: $(actptcpl)",
    "- **middle participle**: $(midptcpl)",
    "- **passive participle**: $(passptcpl)"
]
Markdown.parse(join(mdlines,"\n"))
```

Aorist tense

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("aorist"), gmpVoice("active"),
διδωμι, kds))
```

Middle voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
```

```
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("aorist"), gmpVoice("middle"),
διδωμι, kds))
```

Passive voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("aorist"), gmpVoice("passive"),
διδωμι, kds))
```

Imperative

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")

tbl = md_imperativeconjugation(gmpTense("aorist"), gmpVoice("active"),
διδωμι, kds)
Markdown.parse(tbl)
```

Middle voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")

tbl = md_imperativeconjugation(gmpTense("aorist"), gmpVoice("middle"),
διδωμι, kds)
Markdown.parse(tbl)
```

Passive voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")

tbl = md_imperativeconjugation(gmpTense("aorist"), gmpVoice("passive"),
διδωμι, kds)
Markdown.parse(tbl)
```

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")



inf_act = GMFInfinitive(
    gmpTense("aorist"), gmpVoice("active")
)
inf_actforms = generate(διδωμι, formurn(inf_act), kds)


inf_mdl = GMFInfinitive(
    gmpTense("aorist"), gmpVoice("middle")
)
inf_mdlforms = generate(διδωμι, formurn(inf_mdl), kds)


inf_pass = GMFInfinitive(
    gmpTense("aorist"), gmpVoice("passive")
)
inf_passforms = generate(διδωμι, formurn(inf_pass), kds)



actptcpl = participleslashline(διδωμι, gmpTense("aorist"),
gmpVoice("active"), kds)

midptcpl = participleslashline(διδωμι, gmpTense("aorist"),
gmpVoice("middle"), kds)
```

```
passptcpl = participleslashline(διδωμι, gmpTense("aorist"),
gmpVoice("passive"), kds)


mdlines = [
    "- **active infinitive**: $(inf_actforms[1])",
    "- **middle infinitive**: $(inf_mdlforms[1])",
    "- **passive infinitive**: $(inf_passforms[1])",
    "- **active participle**: $(actptcpl)",
    "- **middle participle**: $(midptcpl)",
    "- **passive participle**: $(passptcpl)"
]

Markdown.parse(join(mdlines,"\n"))
```

Perfect system

Perfect tense

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("perfect"), gmpVoice("active"),
διδωμι, kds))
```

Middle and passive voices (identical forms):

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("perfect"), gmpVoice("passive"),
διδωμι, kds))
```

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")

actptcpl = participleslashline(διδωμι, gmpTense("perfect"),
gmpVoice("active"), kds)

mpptcpl = participleslashline(διδωμι, gmpTense("perfect"),
gmpVoice("middle"), kds)


inf_act = GMFInfinitive(
    gmpTense("perfect"), gmpVoice("active")
)
inf_actforms = generate(διδωμι, formurn(inf_act), kds)

mdlines = [
    "- **active infinitive**: $(inf_actforms[1])",
    "- **active participle**: $(actptcpl)",
    "- **middle & passive participle**: $(mpptcpl)"
]
Markdown.parse(join(mdlines,"\n"))
```

Pluperfect tense

Active voice:

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
διδωμι = LexemeUrn("lsj.n26447")



Markdown.parse(md_conjugation(gmpTense("pluperfect"), gmpVoice("active"),
διδωμι, kds))
```

Middle and passive voices (identical forms):

```
using Kanones, CitableParserBuilder, Markdown
repo = pwd() |> dirname |> dirname |> dirname |> dirname
src1 = joinpath(repo, "datasets", "literarygreek-rules")
src2 = joinpath(repo, "datasets", "lsj")
kds = Kanones.FilesDataset([src1, src2])
```

```
διδωμι = LexemeUrn("lsj.n26447")


Markdown.parse(md_conjugation(gmpTense("pluperfect"), gmpVoice("passive"), διδωμι, kds))
```