SER 321 A Session

SI Session

Thursday, January 23rd 2025

7:00 pm - 8:00 pm MST

Agenda

- URL components
- HTTP
- JSON
  - Overview
  - Hands-on Practice

SI Session Expectations

Thanks for coming to the **SER 321** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
  - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features

Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

Given the following URL, identify the…

Query

https://www.google.com/search?q=asu

Given the following URL, identify the…

Protocol

https://www.google.com/search?q=asu

Given the following URL, identify the…

Path

https://www.google.com/search?q=asu

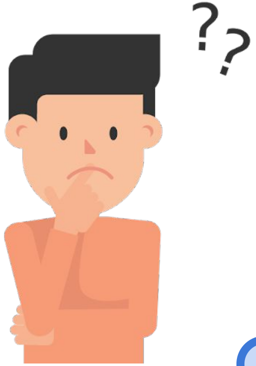
Given the following URL, identify the…

Host

https://www.google.com/search?q=asu

SER 321

HTTP(S)



Stateful

OR

Stateless

Synchronous

OR

Asynchronous

MDN HTTP Request Methods

SER 321

HTTP Requests

We will focus on ***FOUR*** request types

1.

2.

3.

4.

MDN HTTP Request Methods

What’s the difference?

1. GET
2. POST
3. PUT
4. DELETE

Idempotent

SER 321

HTTP Responses

Status Codes →

- 1XX
- 2XX
- 3XX
- 4XX
- 5XX

SER 321

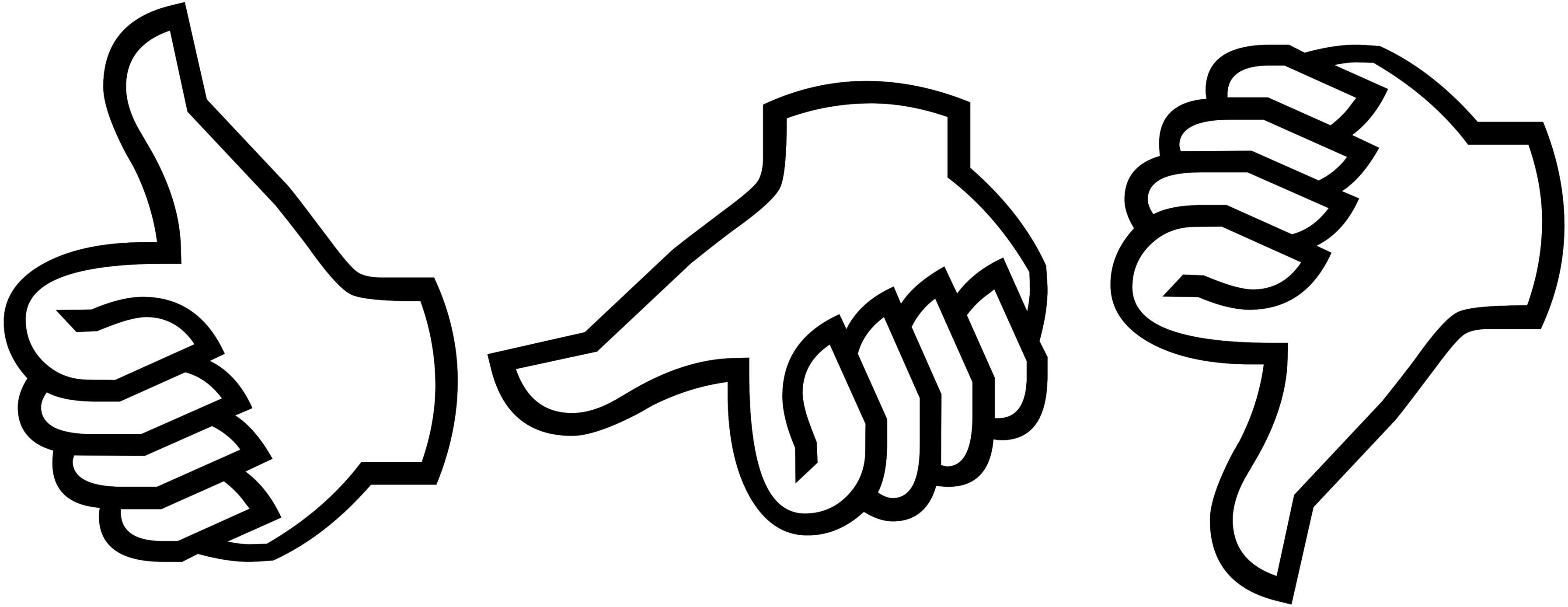
HTTP Responses

1XX	Information	Rare			
2XX	Success	200	201	204	
3XX	Redirect	301			
4XX	User Error	400	403	404	405
5XX	Server Error	501	503		

[org.json Docs]

[JSON Guide]

How comfortable are we with JSON?



org.json Docs

JSON Guide

SER 321

JSON

Object { }

Array [ ]

Member “name” : value

org.json Docs

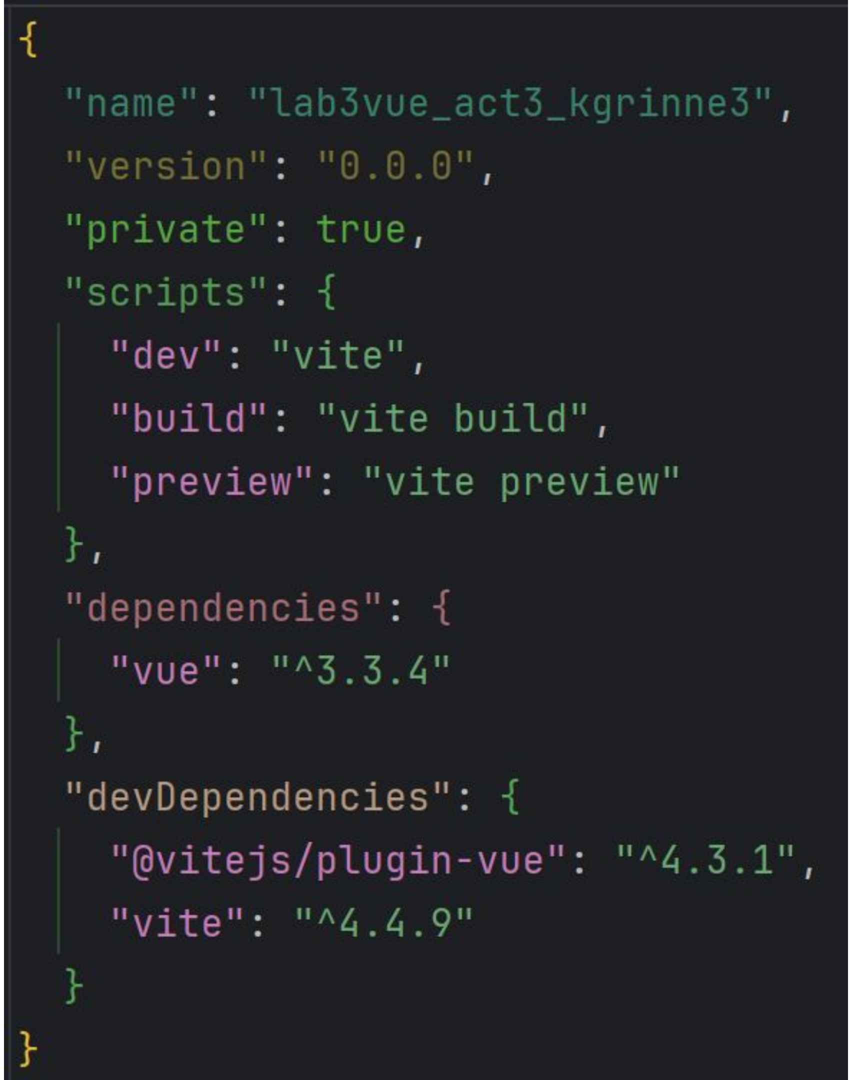
SER 321

JSON

How many Objects?

How many Arrays?

How many Members?



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

JSON Guide

What do we see?

1.

```
{
        “name” : “katie”,
        “role” : “student”,
        “course” : “ser321”
}
```

2.

```
[
    {“name” : “katie”}
]
```

3.

```
{
        “name” : “katie”,
        “pets” : {
                “dog” : “smokey”
                }
}
```

4.

```
{
        “submissions” : [
                {“name” : “katie”},
                {“name” : “zac”}
        ]
}
```

org.json Docs

JSON Guide

SER 321

JSON

Do you see any errors?

1.
```
{
        “name” : “katie”,
        “role” : “student”,
        “course” : “ser321”
}
```

2.
```
[{
        “name” : “katie”,
        “name” : “zac”
}]
```

3.
```
{
        “name” : “katie”,
        “pets” : {
            “dog” : “smokey”,
            “dog” : “samie”
            }
}
```

4.
```
{
        “submissions” : [
                {“name” : “katie”},
                {“name” : “zac”}
        ]
}
```

org.json Docs

JSON Guide

SER 321

JSON

Can we fix these?

```
[
        {“name” : “katie”},
        {“name” : “zac”}
]
```

Correct!

```
[{
        “name” : “katie”,
        “name” : “zac”
}]
```

WRONG

Can we fix these?

```
[
        {“name” : “katie”},
        {“name” : “zac”}
]
```

Correct!

```
[{
        “name” : “katie”,
        “name” : “zac”
}]
```

WRONG

```
{
        “name” : “katie”,
        “pets” : {
                “dog” : “smokey”,
                “dog” : “samie”
                }
}
```

WRONG

```
{
        “name” : “katie”,
        “pets” : {
                {“dog” : “smokey”},
                {“dog” : “samie”}
                }
}
```

Correct!

org.json Docs

JSON Guide

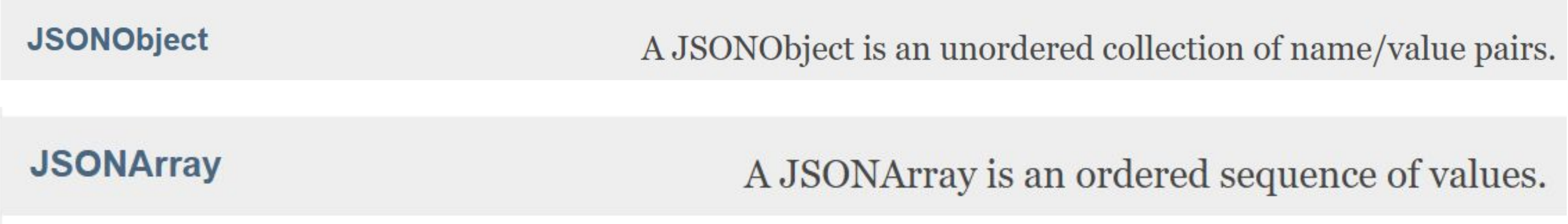
SER 321

JSON in Assignments

Starter Code files use **org.json**

If you have a different JSON library you would prefer to use, you *can* use that

This library is ***very*** intuitive, so I recommend using it!

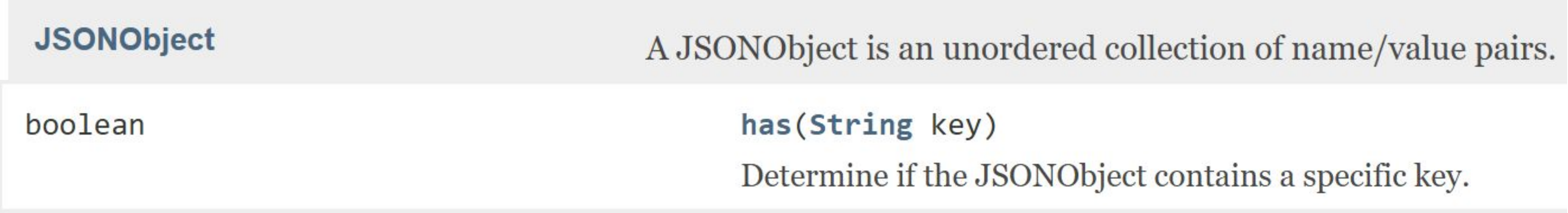


JSONObject	A JSONObject is an unordered collection of name/value pairs.
JSONArray	A JSONArray is an ordered sequence of values.

org.json Docs JSON Guide

SER 321
JSON in Assignments

Very Similar to a **MAP** object



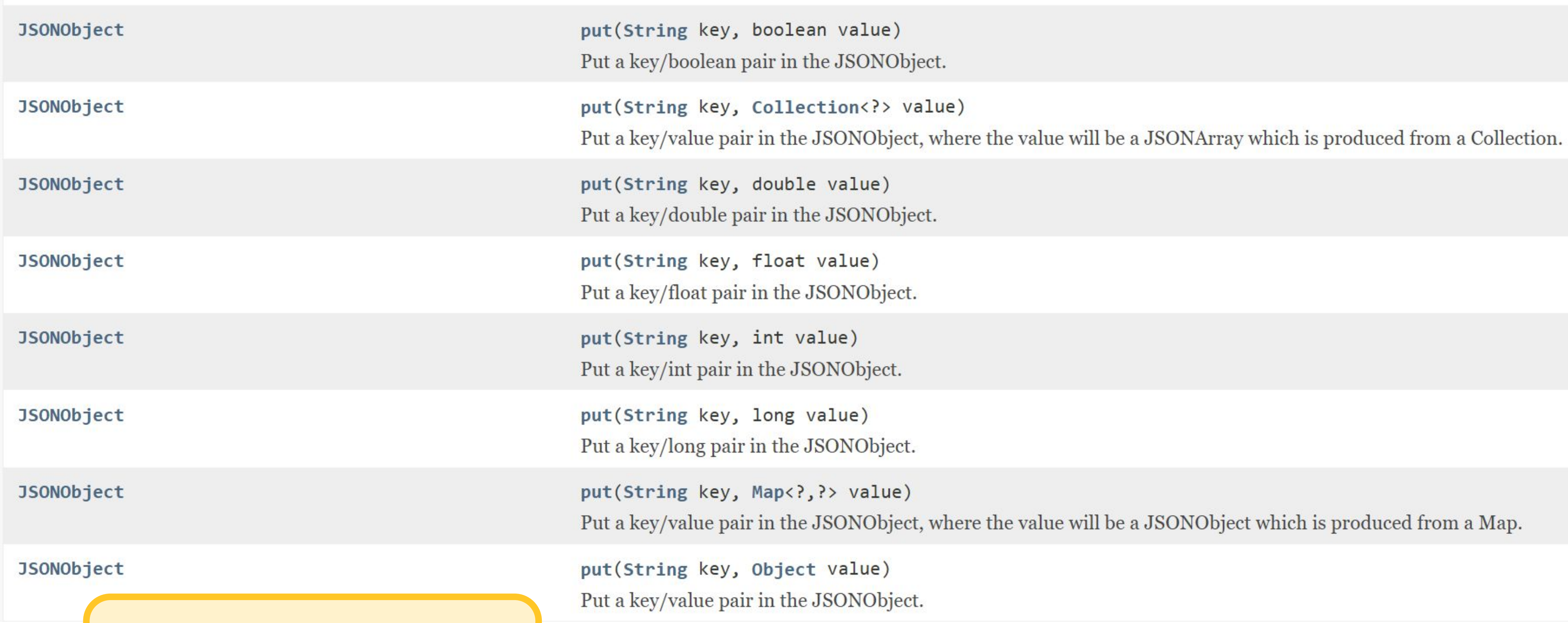
JSONObject	A JSONObject is an unordered collection of name/value pairs.
boolean	has(String key) Determine if the JSONObject contains a specific key.

```
boolean hasStu = json.has(“student”);
```

org.json Docs

JSON Guide

SER 321



`JSONObject`	`put(String key, boolean value)` Put a key/boolean pair in the JSONObject.
`JSONObject`	`put(String key, Collection<?> value)` Put a key/value pair in the JSONObject, where the value will be a JSONArray which is produced from a Collection.
`JSONObject`	`put(String key, double value)` Put a key/double pair in the JSONObject.
`JSONObject`	`put(String key, float value)` Put a key/float pair in the JSONObject.
`JSONObject`	`put(String key, int value)` Put a key/int pair in the JSONObject.
`JSONObject`	`put(String key, long value)` Put a key/long pair in the JSONObject.
`JSONObject`	`put(String key, Map<?,?> value)` Put a key/value pair in the JSONObject, where the value will be a JSONObject which is produced from a Map.
`JSONObject`	`put(String key, Object value)` Put a key/value pair in the JSONObject.

Put will add to the Object

```
json.put(“term”, “spring”);
```

org.json Docs JSON Guide

SER 321

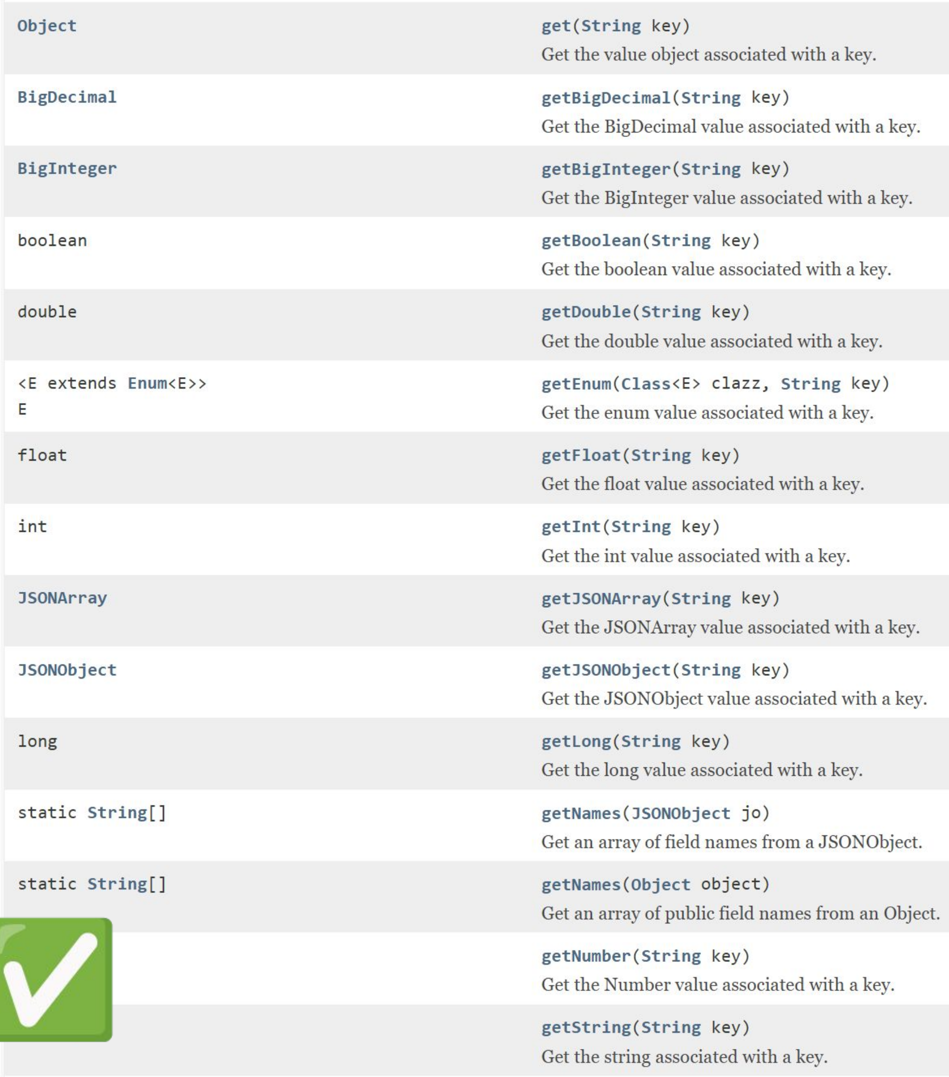
JSON in Assignments

Need to know the data type when pulling from the JSONObject!

```
JSONObject json = {
         “Student” : “katie”,
         “term” : “fall”
}
```

```
String term = json.get(“term”);
```
🚫

```
String term = json.getString(“term”);
```
✅



Object	get(String key) Get the value object associated with a key.
BigDecimal	getBigDecimal(String key) Get the BigDecimal value associated with a key.
BigInteger	getBigInteger(String key) Get the BigInteger value associated with a key.
boolean	getBoolean(String key) Get the boolean value associated with a key.
double	getDouble(String key) Get the double value associated with a key.
<E extends Enum<E>> E	getEnum(Class<E> clazz, String key) Get the enum value associated with a key.
float	getFloat(String key) Get the float value associated with a key.
int	getInt(String key) Get the int value associated with a key.
JSONArray	getJSONArray(String key) Get the JSONArray value associated with a key.
JSONObject	getJSONObject(String key) Get the JSONObject value associated with a key.
long	getLong(String key) Get the long value associated with a key.
static String[]	getNames(JSONObject jo) Get an array of field names from a JSONObject.
static String[]	getNames(Object object) Get an array of public field names from an Object.
	getNumber(String key) Get the Number value associated with a key.
	getString(String key) Get the string associated with a key.

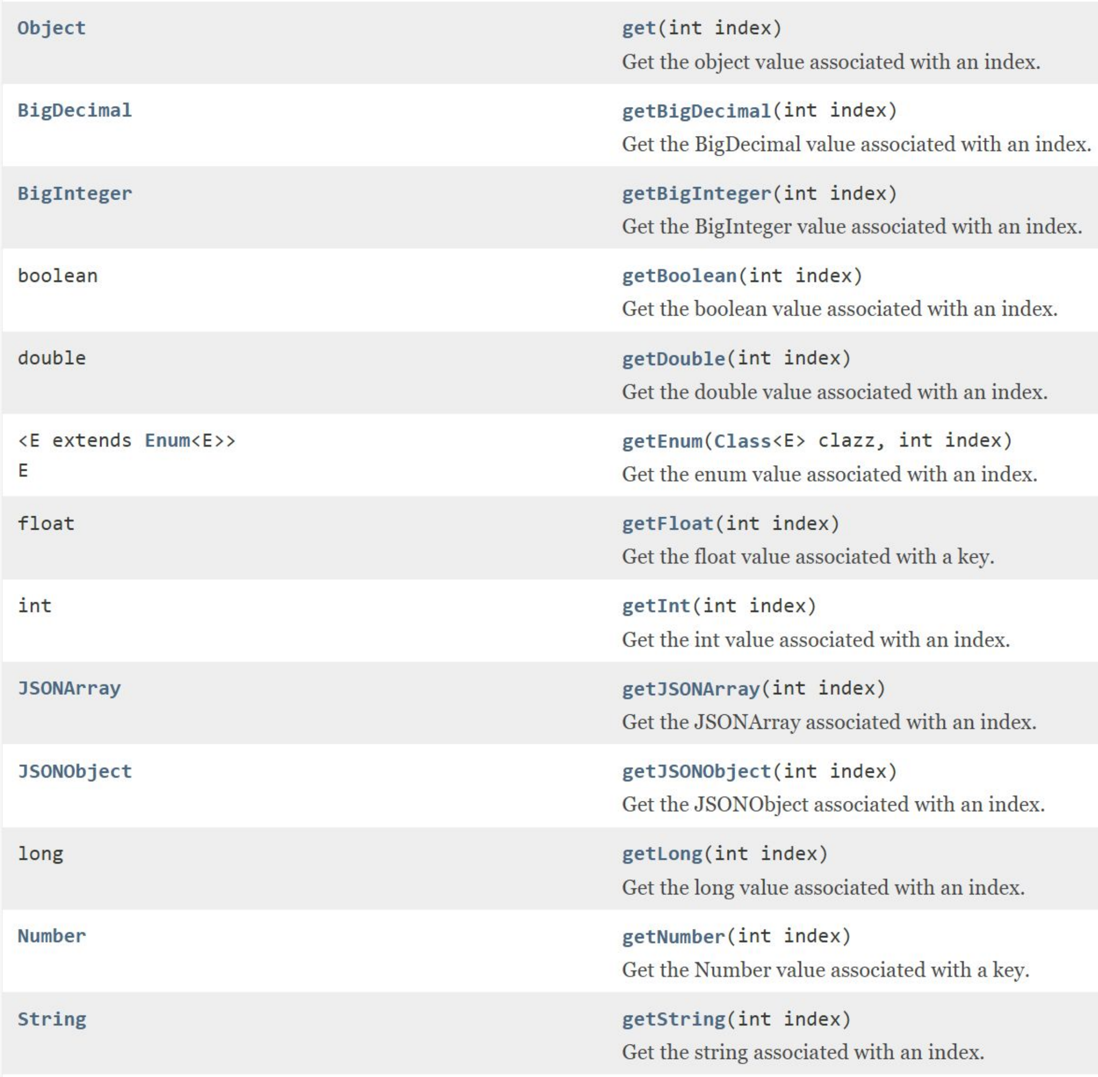
org.json Docs JSON Guide

SER 321
JSON in Assignments

JSONArray

No HAS

Can target indices



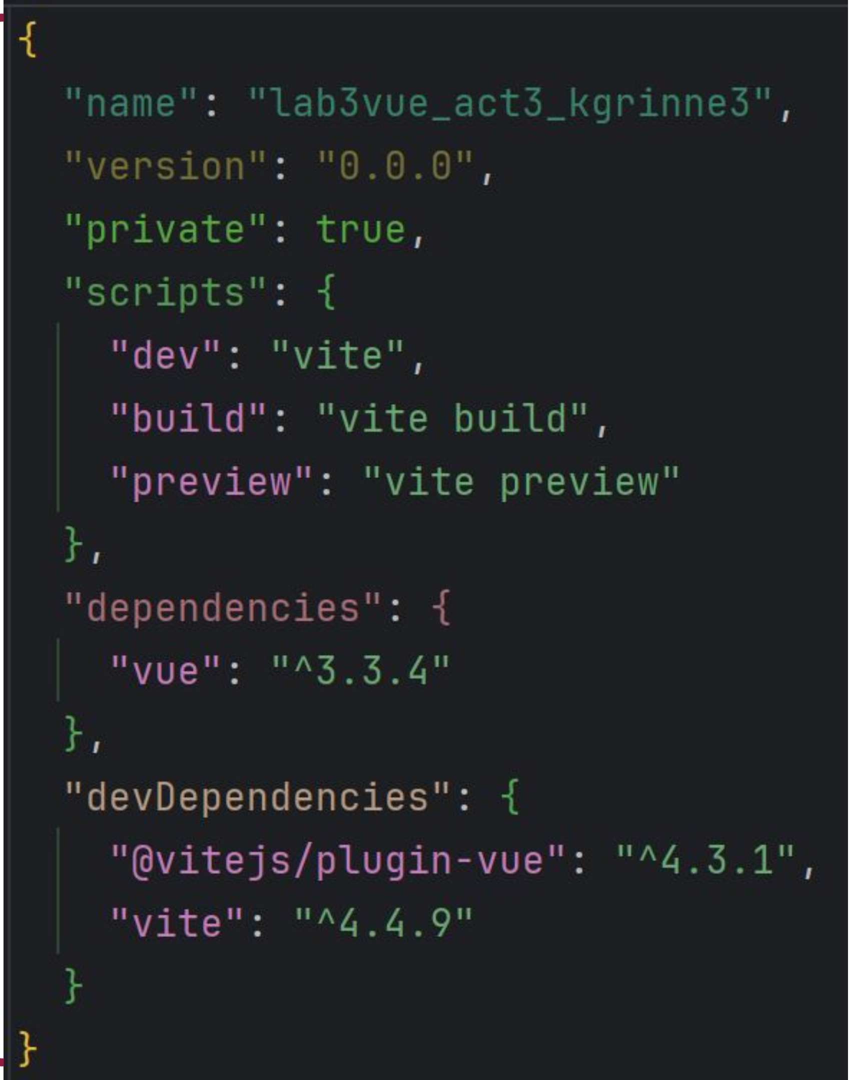
Object	get(int index) Get the object value associated with an index.
BigDecimal	getBigDecimal(int index) Get the BigDecimal value associated with an index.
BigInteger	getBigInteger(int index) Get the BigInteger value associated with an index.
boolean	getBoolean(int index) Get the boolean value associated with an index.
double	getDouble(int index) Get the double value associated with an index.
<E extends Enum<E>> E	getEnum(Class<E> clazz, int index) Get the enum value associated with an index.
float	getFloat(int index) Get the float value associated with a key.
int	getInt(int index) Get the int value associated with an index.
JSONArray	getJSONArray(int index) Get the JSONArray associated with an index.
JSONObject	getJSONObject(int index) Get the JSONObject associated with an index.
long	getLong(int index) Get the long value associated with an index.
Number	getNumber(int index) Get the Number value associated with a key.
String	getString(int index) Get the string associated with an index.

org.json Docs

SER 321

JSON Practice

```
JSONObject json =
```



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

JSON Guide

How would we…

Check for the name member?

```
boolean hasName =
```

Get the project name?

```
String name =
```

org.json Docs

SER 321

JSON Practice

```
JSONObject json =
```

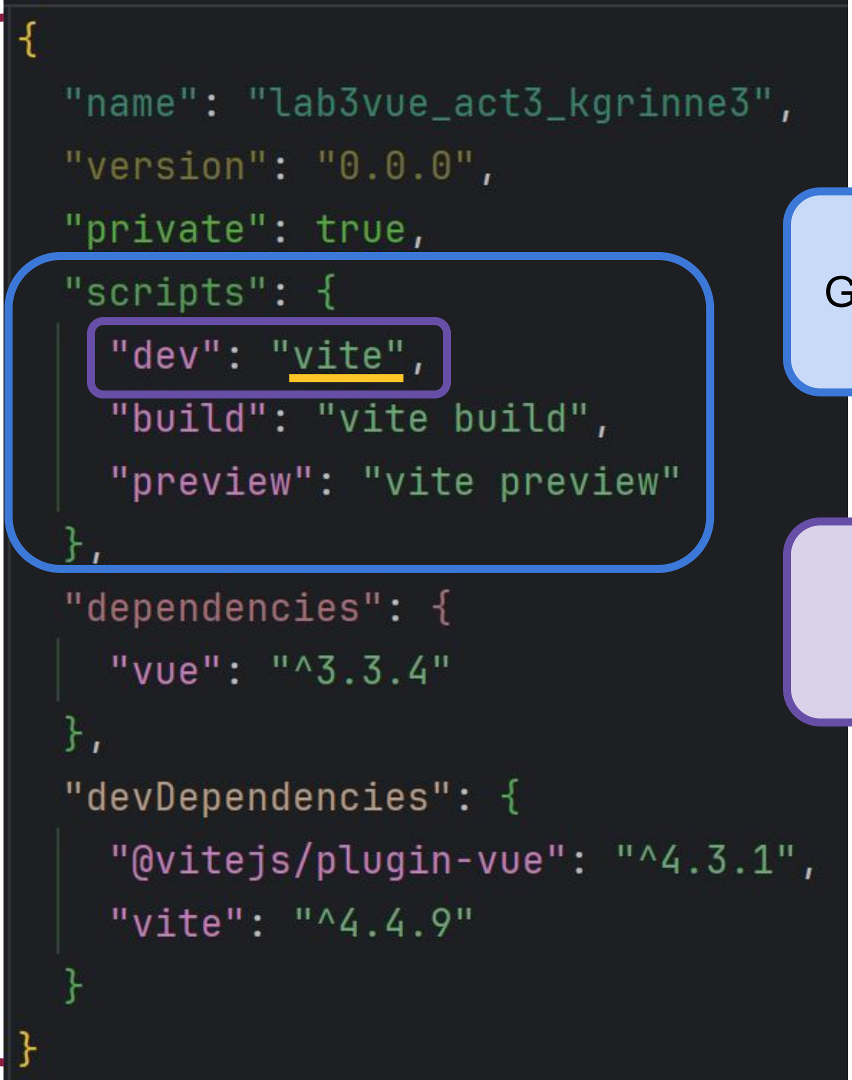
How would we…

Get the dev value?

```
JSONObject scripts =



String dev=
```



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

JSON Guide

Step 1:
Get the `scripts` Object

Step 2:
Get the dev value

org.json Docs

JSON Guide

SER 321

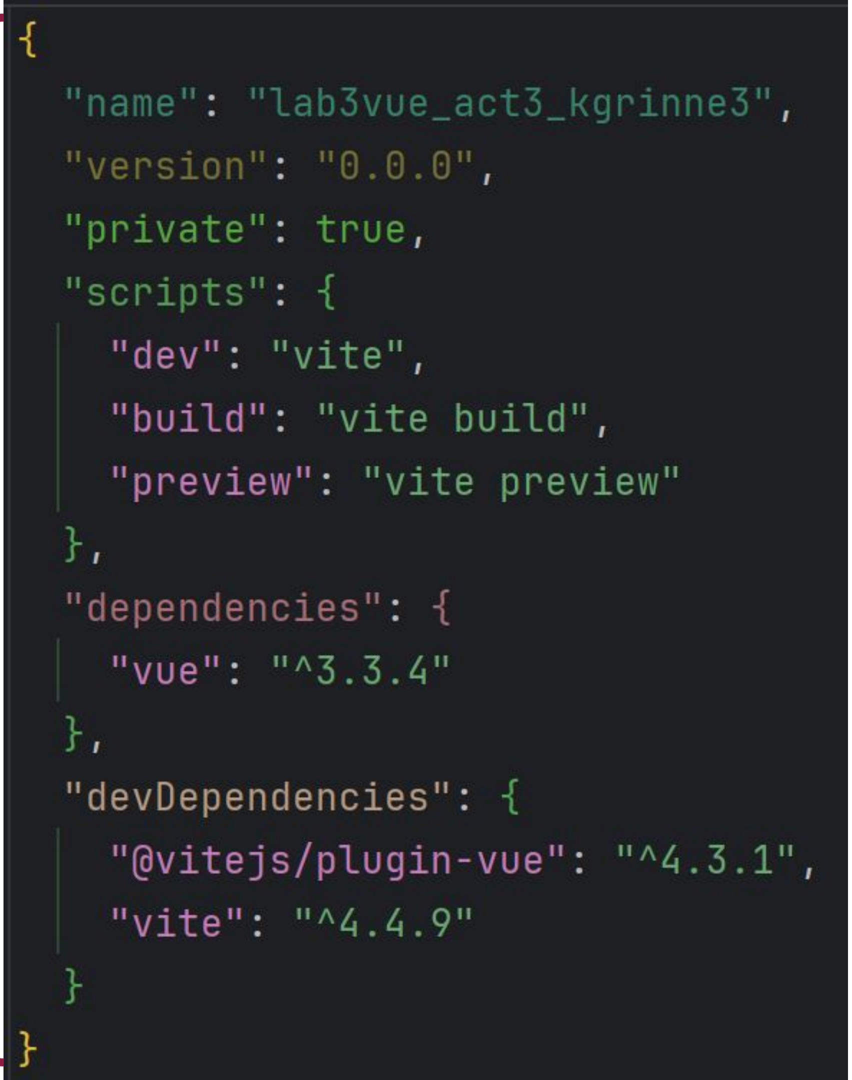
JSON Practice

```
JSONObject json =
```

How would we…

Build the json object?

You *could* do it all inline, but let's go one object at a time for clarity



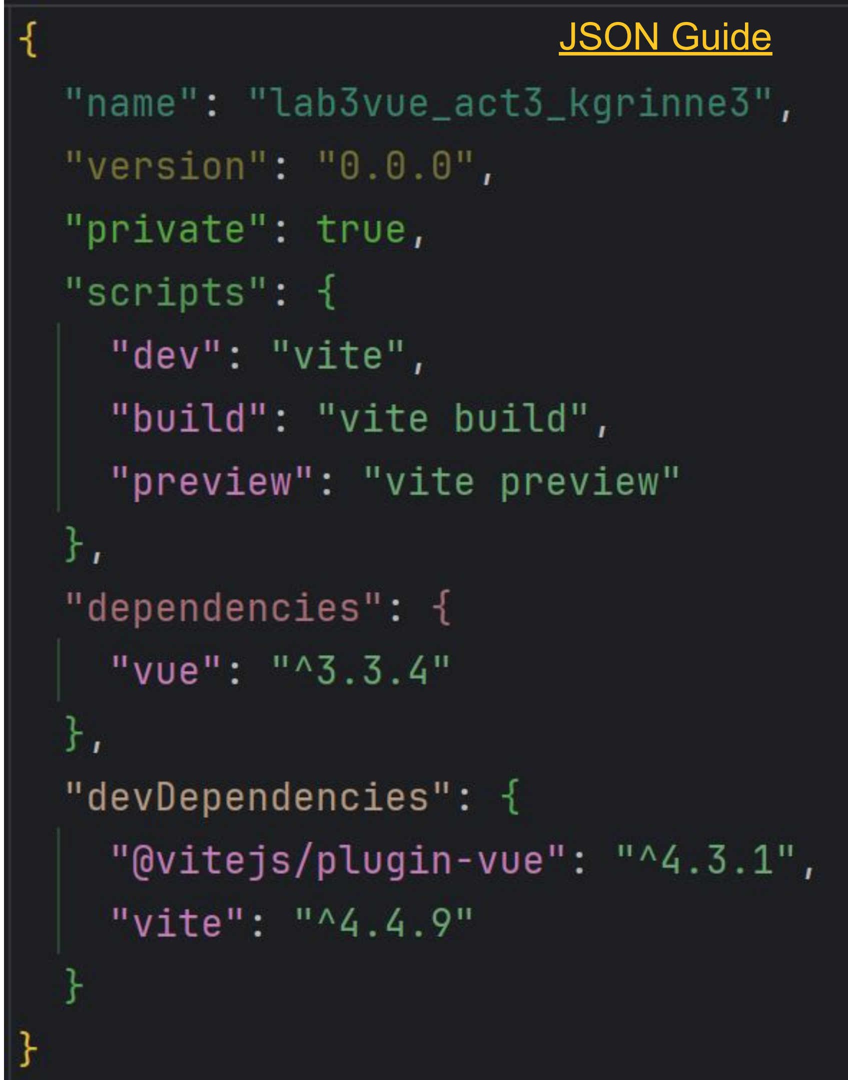
```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

org.json Docs

JSON Guide

SER 321
JSON Practice

```
JSONObject json =
```



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

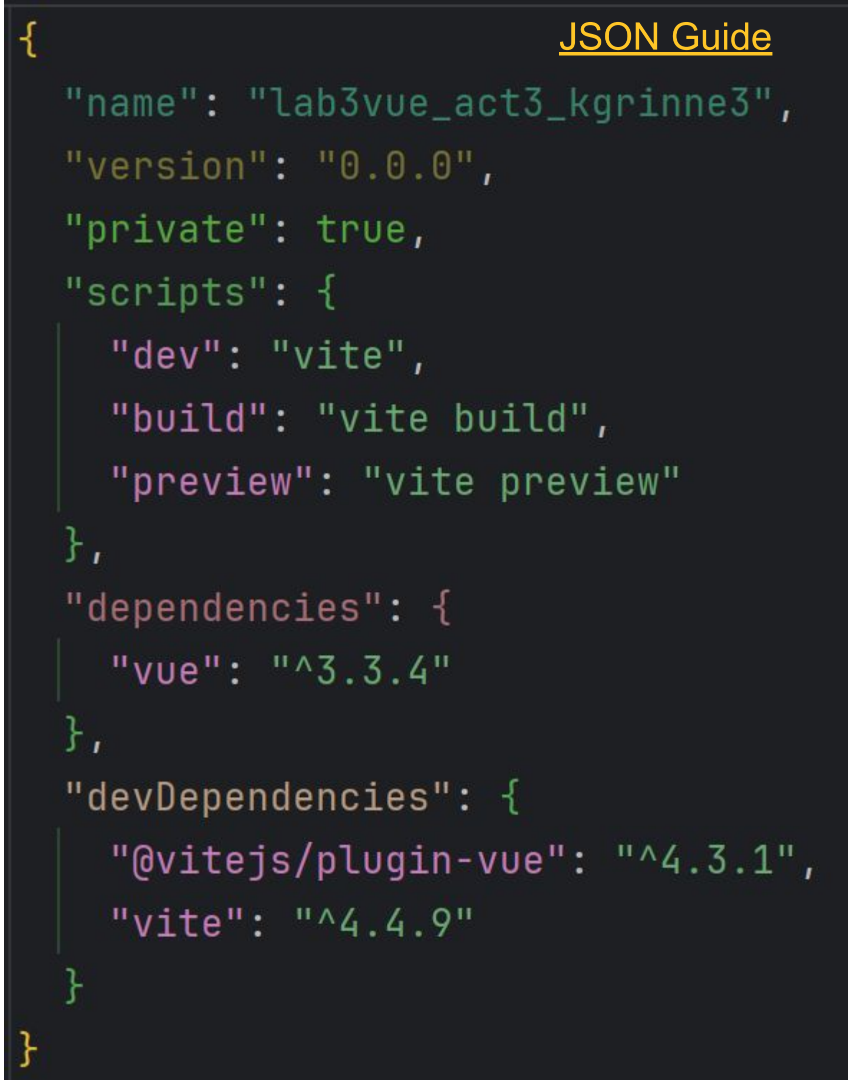
org.json Docs

SER 321

JSON Practice

```
JSONObject json = new JSONObject();
json.put(“name”, “lab3vue_act3_kgrinne3”);
json.put(“version”, “0.0.0”);
json.put(“private”, “true”);
```

JSON Guide



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

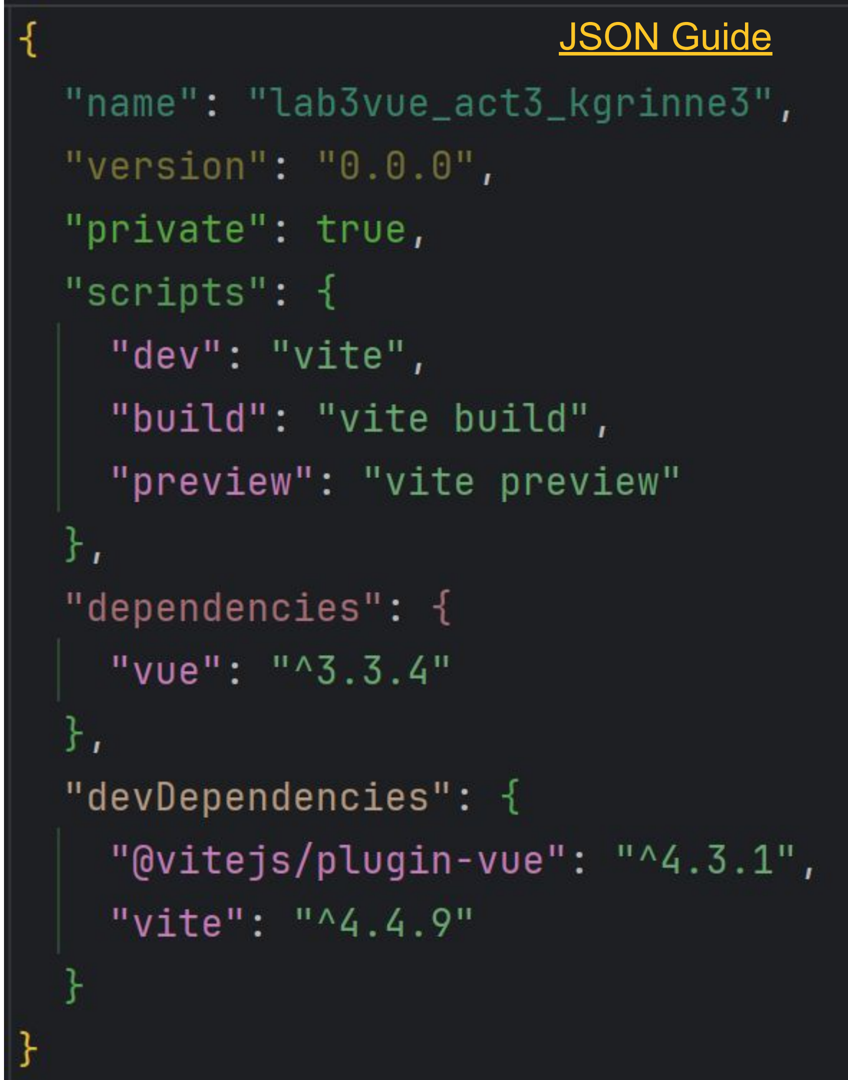
org.json Docs

SER 321

JSON Practice

```
JSONObject scripts =
```

JSON Guide



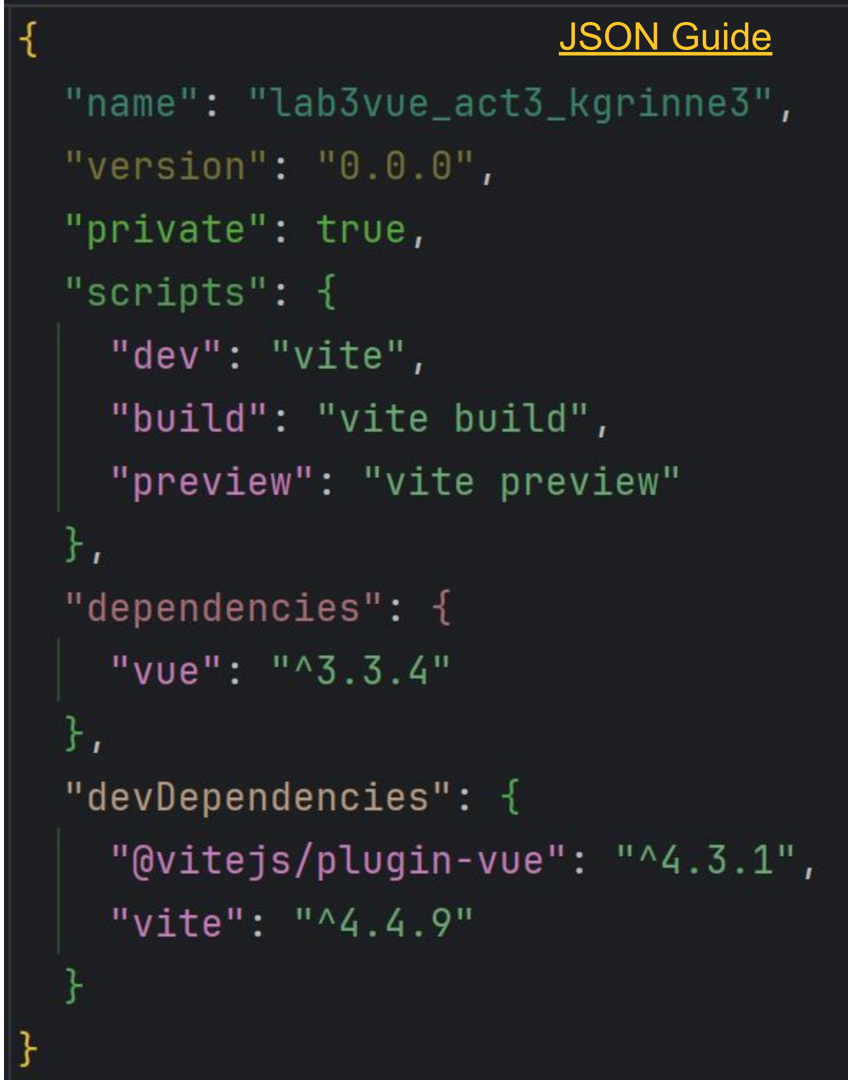
```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

org.json Docs

SER 321

JSON Practice

```
JSONObject scripts = new JSONObject();
scripts.put(“dev”, “vite”);
scripts.put(“build”, “vite build”);
scripts.put(“preview”, “vite preview”);
json.put(“scripts”, scripts.toMap());
```



JSON Guide

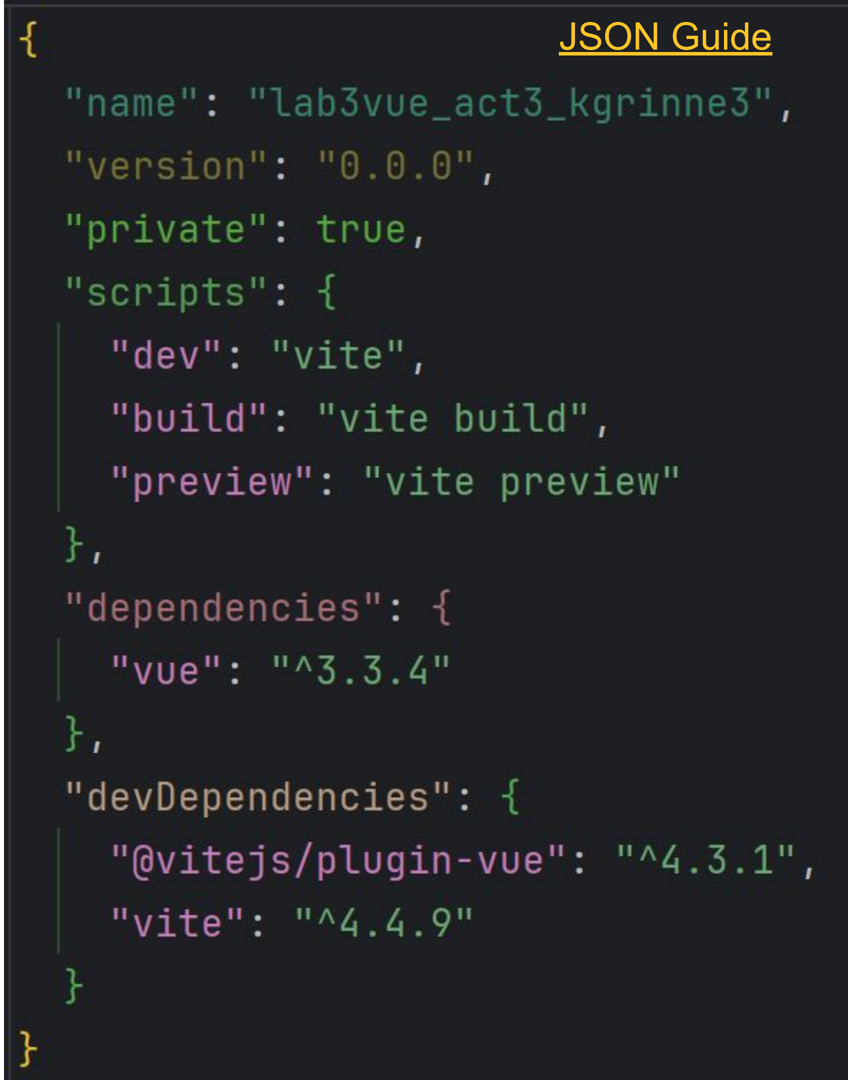
```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

org.json Docs

SER 321

JSON Practice

```
JSONObject depend =
```



JSON Guide

```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

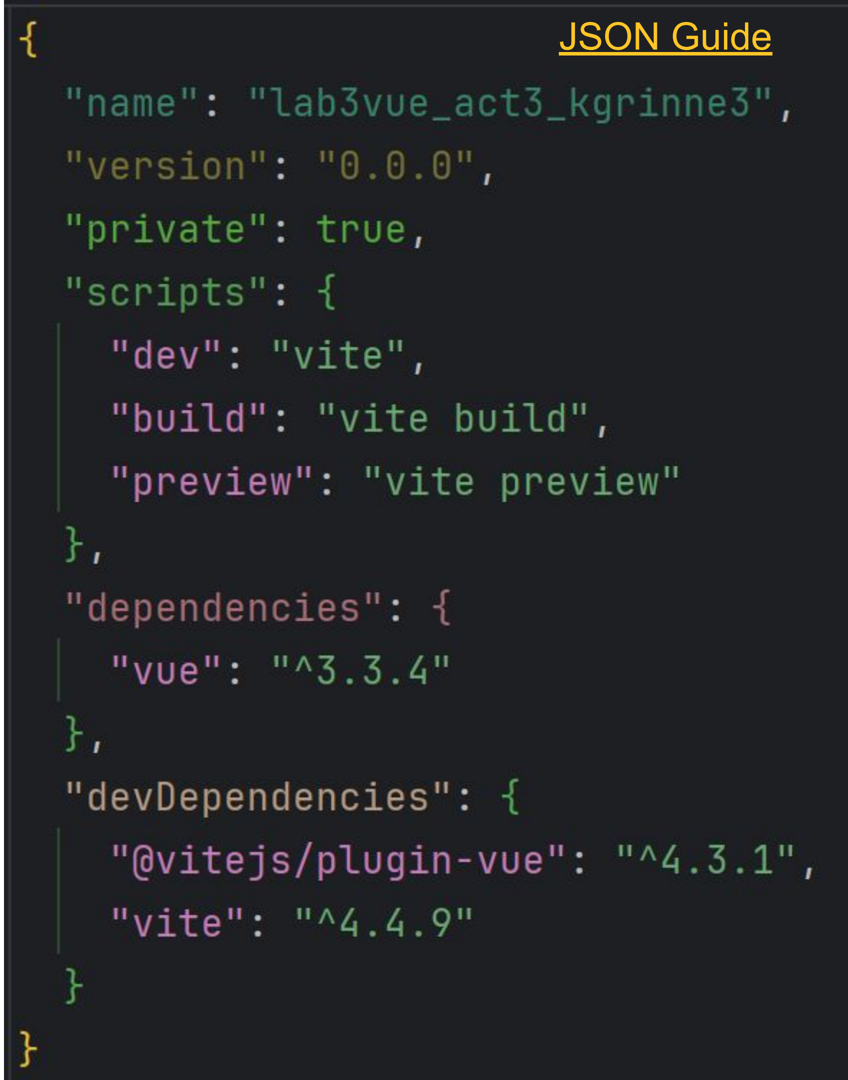
org.json Docs

SER 321

JSON Practice

```
JSONObject depend = new JSONObject();
depend.put("vue", "^3.3.4");
json.put("dependencies", depend.toMap());
```

JSON Guide



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

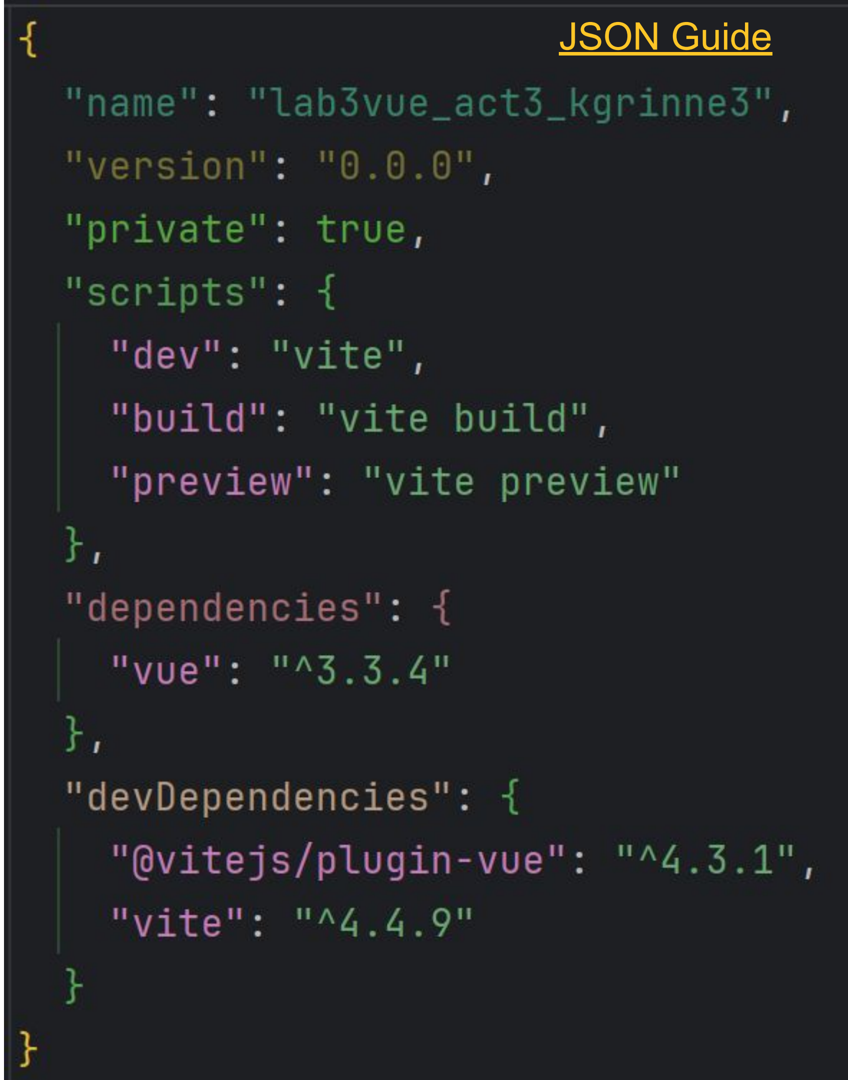
org.json Docs

SER 321

JSON Practice

```
JSONObject devDep =
```

JSON Guide



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

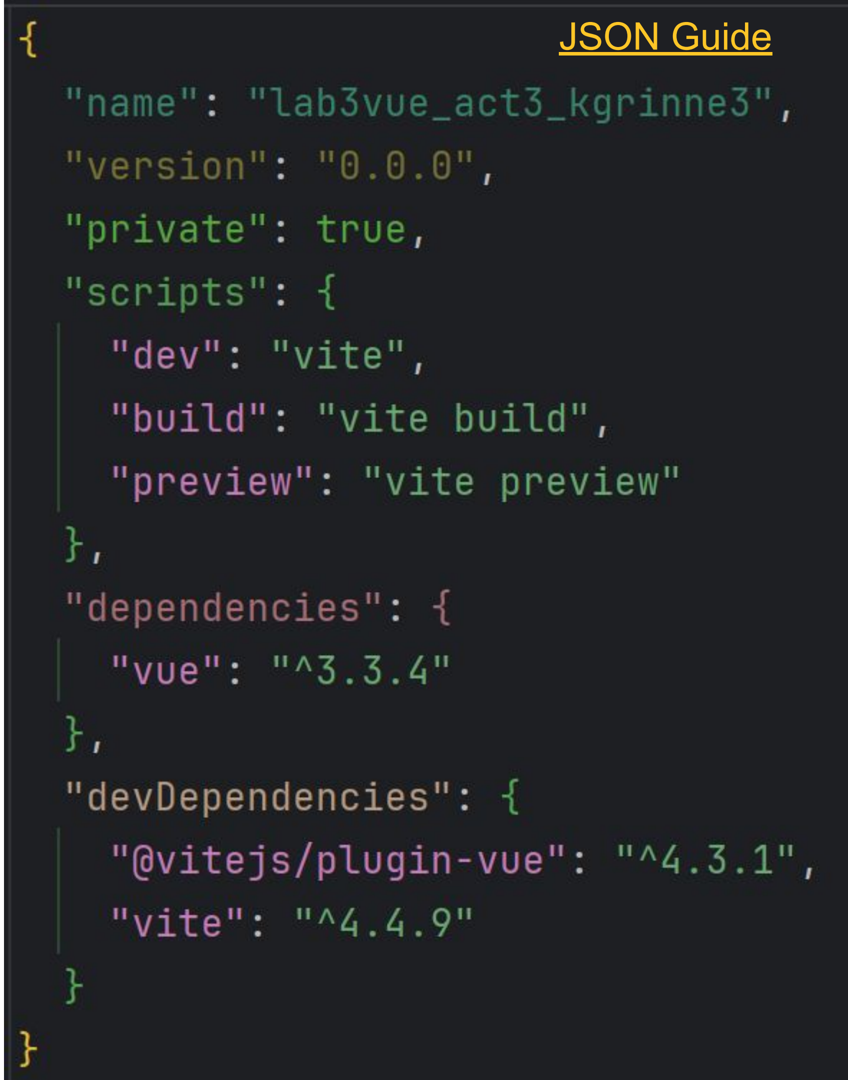
org.json Docs

SER 321

JSON Practice

```
JSONObject devDep = new JSONObject();
devDep.put(“@vitejs/plugin-vue”, “^4.3.1”);
devDep.put(“vite”, “^4.4.9”);
json.put(“devDependencies”, devDep.toMap());
```

JSON Guide



```
{
  "name": "lab3vue_act3_kgrinne3",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "vue": "^3.3.4"
  },
  "devDependencies": {
    "@vitejs/plugin-vue": "^4.3.1",
    "vite": "^4.4.9"
  }
}
```

Connections!

SER 321

Scratch Space

Upcoming Events

SI Sessions:

- Sunday, January 26th at 7:00 pm MST
- Tuesday, January 28th at 11:00 am MST
- Thursday, January 30th at 7:00 pm MST

Review Sessions:

- TBD

Questions?

Survey:

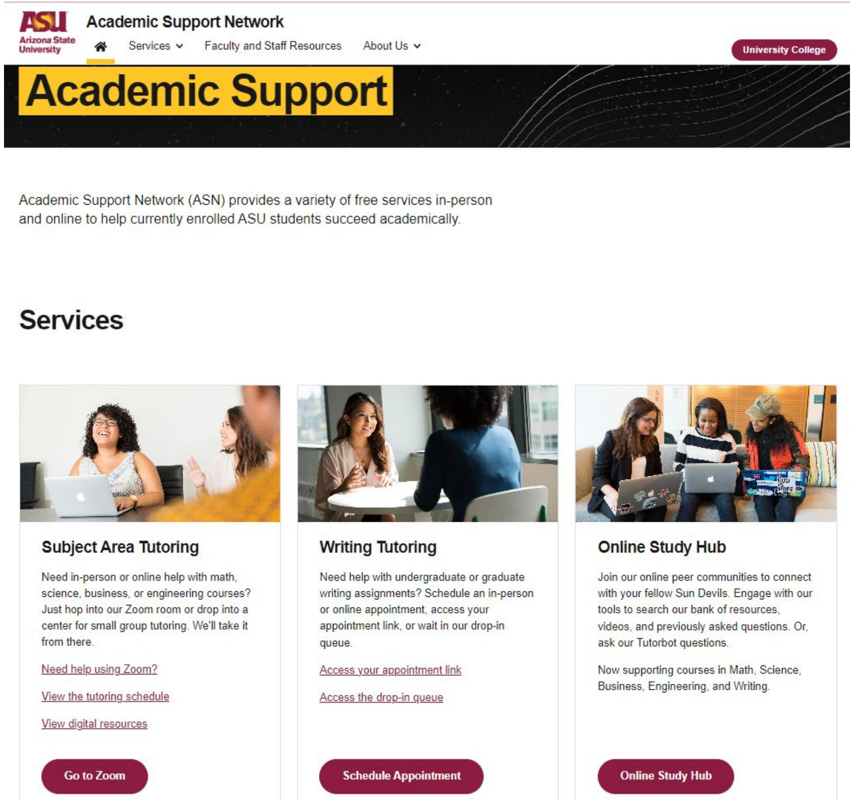
https://asuasn.info/ASNSurvey

40

More Questions?

Check out our other resources!

tutoring.asu.edu



Academic Support Network

Services ⌄ Faculty and Staff Resources About Us ⌄

University College

Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

Services

Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

Need help using Zoom?

View the tutoring schedule

View digital resources

Go to Zoom

Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

Access your appointment link

Access the drop-in queue

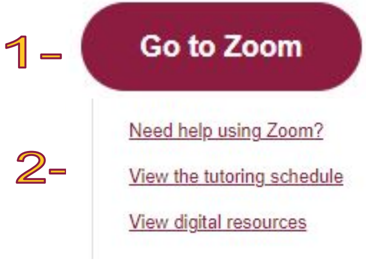
Schedule Appointment

Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub



1 –

Go to Zoom

2-

Need help using Zoom?

View the tutoring schedule

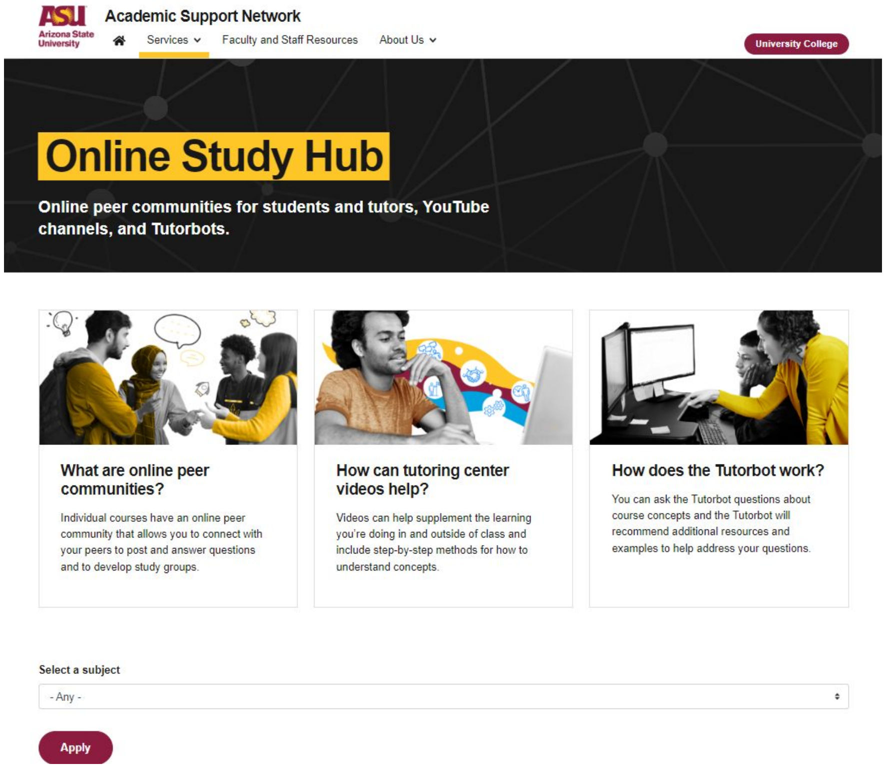
View digital resources

1. **Click on 'Go to Zoom' to log onto our Online Tutoring Center.**
2. **Click on 'View the tutoring schedule' to see when tutors are available for specific courses.**

More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub



Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.

What are online peer communities?

Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.

How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.

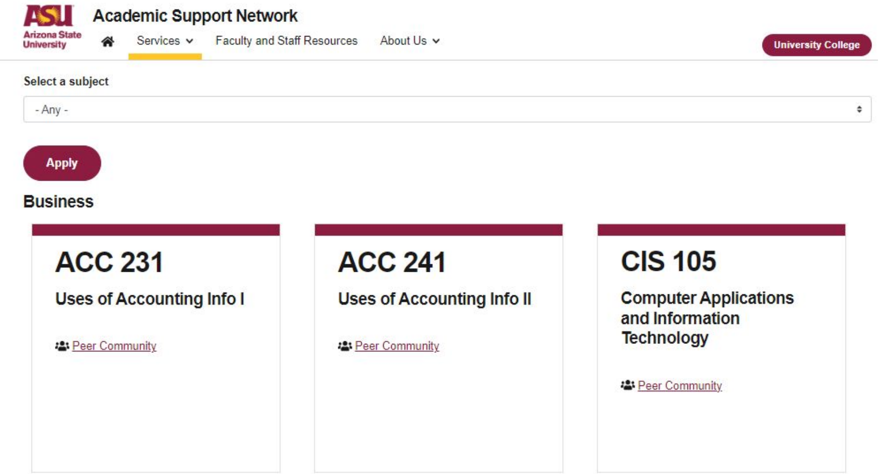
How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.

Select a subject

- Any -

Apply



Select a subject

- Any -

Apply

Business

ACC 231
Uses of Accounting Info I
Peer Community

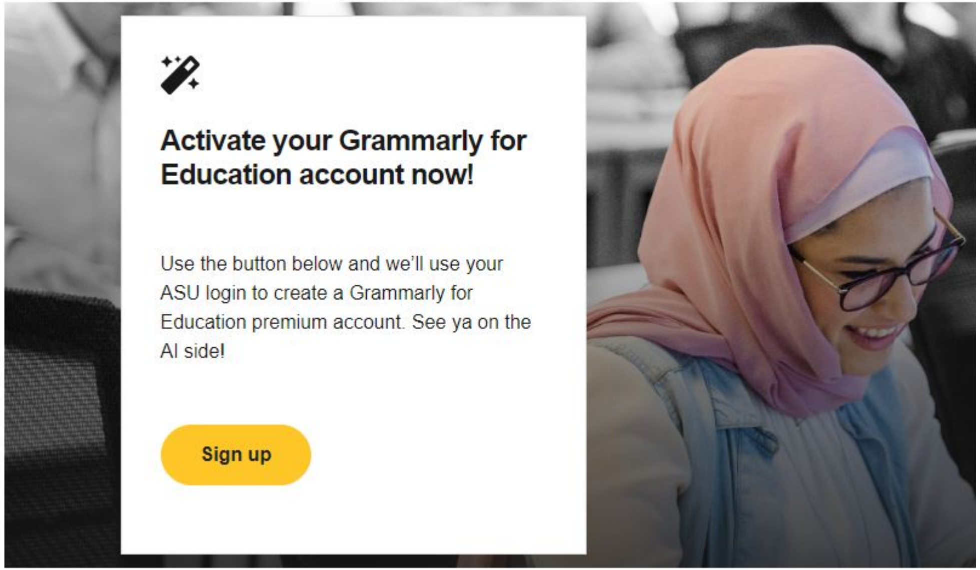
ACC 241
Uses of Accounting Info II
Peer Community

CIS 105
Computer Applications and Information Technology
Peer Community

Don’t forget to check out the Online Study Hub for additional resources!

Expanded Writing Support Available

Including Grammarly for Education, at no cost!



Activate your Grammarly for Education account now!

Use the button below and we'll use your ASU login to create a Grammarly for Education premium account. See ya on the AI side!

tutoring.asu.edu/expanded-writing-support

*Available slots for this pilot are limited

Additional Resources

- Course Repo
- Gradle Documentation
- GitHub SSH Help
- Linux Man Pages
- OSI Interactive
- MDN HTTP Docs
    - Requests
    - Responses
- JSON Guide
- org.json Docs