Certainly! Here are 20 JSON examples covering a range of complexity:

1. Simple JSON object:

```json

{

"key": "value"

}

```

2. JSON object with number, boolean, and null values:

```json

{

"name": "John Doe",

"age": 30,

"isStudent": false,

"address": null

}

```

3. JSON array:

```json

["apple", "banana", "orange"]

```

4. JSON object with an array of objects:

```json

{

"students": [

{

"name": "John Doe",

"age": 20

},

{

"name": "Jane Smith",

"age": 22

}

]

}

```

5. JSON object with nested objects:

```json

{

"person": {

"name": "John Doe",

"address": {

"street": "123 Main St",

"city": "New York"

}

}

}

```

6. JSON object with nested arrays:

```json

{

"employees": [

{

"name": "John Doe",

"skills": ["Java", "Python"]

},

{

"name": "Jane Smith",

"skills": ["C#", "JavaScript"]

}

]

}

```

7. JSON object with date and time:

```json

{

"event": "Meeting",

"date": "2023-05-24",

"time": "10:00:00"

}

```

8. JSON object with string containing special characters:

```json

{

"message": "This is a special character: \\"

}

```

9. JSON object with scientific notation and large number:

```json

{

"pi": 3.14159,

"avogadroNumber": 6.02214e+23

}

```

10. Complex JSON structure combining various features:

```json

{

"user": {

"name": "John Doe",

"age": 30,

"isActive": true,

"emails": ["john@example.com", "johndoe@gmail.com"],

"address": {

"street": "456 Elm St",

"city": "Chicago"

},

"orders": [

{

"id": "123",

"products": ["item1", "item2"],

"quantity": 2

},

{

"id": "456",

"products": ["item3", "item4"],

"quantity": 1

}

]

}

}

```

11. JSON object with boolean and nested objects:

```json

{

"isTrue": true,

"isFalse": false,

"nested": {

"key": "value"

}

}

```

12. JSON array of objects with mixed data types:

```json

[

{

"name": "John Doe",

"age": 30,

"isStudent": false

},

{

"name": "Jane Smith",

"age": 25,

"isStudent": true

}

]

```

13. JSON object with arrays of arrays:

```json

{

"matrix": [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

}

```

14. JSON object with nested arrays of objects:

```json

{

"data": [

{

"name": "John",

"scores": [90, 85, 95]

},

{

"name": "Jane",

"scores": [95, 92, 88]

}

]

}

```

15. JSON object with ISO 8601 formatted date and time:

```json

{

"event": "Meeting",

"timestamp": "2023-05-24T14:30:00Z"

}

```

16. JSON object with nested objects and arrays:

```json

{

"company": "XYZ Corporation",

"departments": [

{

"name": "Engineering",

"employees": [

{

"name": "John Doe",

"skills": ["Java", "Python"]

},

{

"name": "Jane Smith",

"skills": ["C#", "JavaScript"]

}

]

},

{

"name": "Marketing",

"employees": [

{

"name": "Mike Johnson",

"skills": ["Social Media", "Content Creation"]

}

]

}

]

}

```

17. JSON object with nested objects and complex data types:

```json

{

"person": {

"name": "John Doe",

"age": 30,

"birthDate": "1993-07-15",

"address": {

"street": "123 Main St",

"city": "New York"

},

"hobbies": ["reading", "swimming"],

"isEmployed": true,

"salary": 50000.0

}

}

```

18. JSON object with URL and email:

```json

{

"website": "https://www.example.com",

"email": "contact@example.com"

}

```

19. JSON object with conditional values:

```json

{

"name": "John Doe",

"age": 25,

"isAdult": true,

"message": "You are an adult."

}

```

20. JSON object with complex relationships:

```json

{

"users": [

{

"id": 1,

"name": "John Doe",

"orders": [

{

"id": 1001,

"products": [

{

"id": 101,

"name": "Product A",

"price": 10.0

},

{

"id": 102,

"name": "Product B",

"price": 15.0

}

]

},

{

"id": 1002,

"products": [

{

"id": 103,

"name": "Product C",

"price": 20.0

}

]

}

]

},

{

"id": 2,

"name": "Jane Smith",

"orders": [

{

"id": 2001,

"products": [

{

"id": 104,

"name": "Product D",

"price": 25.0

}

]

}

]

}

]

}

```

These examples cover a wide range of JSON structures, including simple objects, arrays, nesting, different data types, complex relationships, dates and times, URLs, conditional values, and more.