# JSON Schema Cheat Sheet

## Terms

- **Instance** - the JSON document being validated or described
- **Schema** - the document containing the description
- **Constraints** - created by adding validation keywords to the schema

## Type validation keyword values

- `string`
- `integer`
- `number` - allows integer or floating point numbers
- `object`
- `array`
- `boolean`
- `null`

The type keyword can be either a string or an array e.g.

```json
{ "type": "number" }
{ "type": ["number", "string"] }
```

## String type keywords

- `minLength` - number
- `maxLength` - number
- `pattern` - regular expression in ECMA 262 syntax
- `format` - date, date-time, uri, email, hostname, ipv4, ipv6, regex

## Integer/Number type keywords

- `multipleOf` - number
- `minimum` - x >= number
- `exclusiveMinimum` - x > number
- `maximum` - x <= number
- `exclusiveMaximum` - x < number

## Object type keywords

- `minProperties` - number
- `maxProperties` - number
- `required` - array of property names
- `properties` - map of property names to schemas
- `patternProperties` - match property names with regex, validate values against a schema
- `additionalProperties` - boolean - are properties not in the schema allowed?
- `dependencies` - map dependencies between properties
- `propertyNames` - validate property names against a schema

## Array type keywords

- `minItems` - number
- `maxItems` - number
- `uniqueItems` - boolean - should all array items be unique?
- `items` - schema or array of schemas to validate array items against
- `additionalItems` - boolean or schema - false to disallow additional items, or schema for additional items
- `contains` - schema - array must contain at least one item that validates against the schema

## Keywords for combining schemas

- `allOf` - array - Must be valid against all of the subschemas
- `anyOf` - array - Must be valid against any of the subschemas
- `oneOf` - array - Must be valid against exactly one of the subschemas
- `not` - object - Must not be valid against the given schema
- `if/then/else` - Allow for conditional validation

This cheat sheet is based on draft-07 of JSON Schema (the most widely supported specification as of 2020-12) and the features implemented by Ajv.

Spotted a mistake? Drop an email to simon@simonplend.co.uk and let me know!

## Sources

- [https://json-schema.org/specification-links.html#draft-7](https://json-schema.org/specification-links.html#draft-7)
- [https://json-schema.org/understanding-json-schema/](https://json-schema.org/understanding-json-schema/)
- [https://ajv.js.org/keywords.html](https://ajv.js.org/keywords.html)