'use client';

import React, { useState } from 'react';

type FeatureInputProps = {

onSubmit: (features: number[]) => void;

isLoading: boolean;

};

type FieldConfig = {

id: string;

label: string;

description: string;

type: 'number' | 'select';

options?: {value: number, label: string}[];

defaultValue: number;

step?: number;

min?: number;

};

type FieldGroup = {

title: string;

fields: FieldConfig[];

};

const FeatureInput: React.FC<FeatureInputProps> = ({ onSubmit, isLoading }) => {

const fieldGroups: FieldGroup[] = [

{

title: "Configuration Parameters",

fields: [

{

id: "joint\_type",

label: "Joint Type (Jt)",

description: "1=干接缝，2=环氧树脂接缝，3=湿接缝，4=整体浇筑接缝",

type: "select" as const,

options: [

{ value: 1, label: "Dry Joint" },

{ value: 2, label: "Epoxy Joint" },

{ value: 3, label: "Wet Joint" },

{ value: 4, label: "Monolithic Joint" }

],

defaultValue: 1

},

{

id: "specimen\_type",

label: "Specimen Type (St)",

description: "1=单接缝(SJ)，2=双接缝(DJ)",

type: "select" as const,

options: [

{ value: 1, label: "Single Joint (SJ)" },

{ value: 2, label: "Double Joint (DJ)" }

],

defaultValue: 1

},

{

id: "key\_number",

label: "Number of Keys (Nk)",

description: "接缝中键槽的数量",

type: "number" as const,

defaultValue: 1,

min: 0,

step: 1

}

]

},

{

title: "Geometric Parameters",

fields: [

{

id: "key\_width",

label: "Key Width (Bk)",

description: "单个键槽的宽度",

type: "number" as const,

defaultValue: 35,

step: 0.1,

min: 0

},

{

id: "key\_root\_height",

label: "Key Root Height (Hk)",

description: "单个键槽根部的高度",

type: "number" as const,

defaultValue: 50,

step: 0.1,

min: 0

},

{

id: "key\_depth",

label: "Key Depth (Dk)",

description: "单个键槽的深度",

type: "number" as const,

defaultValue: 25,

step: 0.1,

min: 0

},

{

id: "key\_inclination",

label: "Key Inclination (theta\_k)",

description: "单个键槽的倾斜角度",

type: "number" as const,

defaultValue: 34.5,

step: 0.1

},

{

id: "key\_spacing",

label: "Key Spacing (Sk)",

description: "相邻键槽之间的距离",

type: "number" as const,

defaultValue: 50,

step: 0.1,

min: 0

},

{

id: "key\_front\_height",

label: "Key Front Height (hk)",

description: "单个键槽前部的高度",

type: "number" as const,

defaultValue: 25,

step: 0.1,

min: 0

},

{

id: "key\_depth\_height\_ratio",

label: "Key Depth-Height Ratio (Dk/Hk)",

description: "键槽深度与高度的比值",

type: "number" as const,

defaultValue: 0.5,

step: 0.01,

min: 0

},

{

id: "joint\_width",

label: "Joint Width (Bj)",

description: "接缝的总宽度",

type: "number" as const,

defaultValue: 200,

step: 0.1,

min: 0

},

{

id: "joint\_height",

label: "Joint Height (Hj)",

description: "接缝的总高度",

type: "number" as const,

defaultValue: 200,

step: 0.1,

min: 0

},

{

id: "key\_area",

label: "Key Area (Ak)",

description: "接缝中键槽区域的总面积",

type: "number" as const,

defaultValue: 17500,

step: 0.1,

min: 0

},

{

id: "joint\_area",

label: "Joint Area (Aj)",

description: "接缝的总面积",

type: "number" as const,

defaultValue: 40000,

step: 0.1,

min: 0

},

{

id: "flat\_region\_area",

label: "Flat Region Area (Asm)",

description: "接缝中平坦区域的面积",

type: "number" as const,

defaultValue: 38250,

step: 0.1,

min: 0

},

{

id: "key\_joint\_area\_ratio",

label: "Key-Joint Area Ratio (Ak/Aj)",

description: "键槽面积与接缝面积的比值",

type: "number" as const,

defaultValue: 0.04,

step: 0.01,

min: 0

}

]

},

{

title: "UHPC Material Properties",

fields: [

{

id: "compressive\_strength",

label: "Compressive Strength (fc)",

description: "UHPC材料的抗压强度",

type: "number" as const,

defaultValue: 193,

step: 0.1,

min: 0

},

{

id: "fiber\_type",

label: "Fiber Type (Ft)",

description: "0=无纤维，1=直纤维，2=混合纤维(直纤维和端钩纤维)，3=端钩纤维",

type: "select" as const,

options: [

{ value: 0, label: "No Fiber" },

{ value: 1, label: "Straight Fiber" },

{ value: 2, label: "Mixed Fiber (Straight and Hooked)" },

{ value: 3, label: "Hooked Fiber" }

],

defaultValue: 1

},

{

id: "fiber\_volume\_fraction",

label: "Fiber Volume Fraction (pf)",

description: "纤维在UHPC中的体积分数",

type: "number" as const,

defaultValue: 0.01,

step: 0.01,

min: 0

},

{

id: "fiber\_length",

label: "Fiber Length (lf)",

description: "纤维的长度",

type: "number" as const,

defaultValue: 13,

step: 0.1,

min: 0

},

{

id: "fiber\_diameter",

label: "Fiber Diameter (df)",

description: "纤维的直径",

type: "number" as const,

defaultValue: 0.2,

step: 0.01,

min: 0

},

{

id: "fiber\_reinforcing\_index",

label: "Fiber Reinforcing Index (lambda\_f)",

description: "纤维增强指数，计算公式为 pf×lf/df",

type: "number" as const,

defaultValue: 65,

step: 0.01,

min: 0

}

]

},

{

title: "Confinement Stress Parameters",

fields: [

{

id: "confining\_stress",

label: "Confinement Stress (sigma\_n)",

description: "约束应力值",

type: "number" as const,

defaultValue: 1,

step: 0.1,

min: 0

},

{

id: "confining\_ratio",

label: "Confinement Ratio (sigma\_n/fc)",

description: "约束应力与抗压强度之比",

type: "number" as const,

defaultValue: 0.005,

step: 0.001,

min: 0

}

]

},

{

title: "Shear Strength",

fields: [

{

id: "shear\_strength",

label: "Shear Strength (tau\_max)",

description: "接缝的最大抗剪强度",

type: "number" as const,

defaultValue: 1421,

step: 0.1,

min: 0

}

]

}

];

const initialValues: Record<string, number> = {};

fieldGroups.forEach(group => {

group.fields.forEach(field => {

initialValues[field.id] = field.defaultValue;

});

});

const [values, setValues] = useState<Record<string, number>>(initialValues);

const [expanded, setExpanded] = useState<Record<string, boolean>>({

"Configuration Parameters": true,

"Geometric Parameters": true,

"UHPC Material Properties": true,

"Confinement Stress Parameters": true,

"Shear Strength": true

});

const handleChange = (id: string, value: number) => {

setValues(prev => ({

...prev,

[id]: value

}));

};

const toggleGroup = (title: string) => {

setExpanded(prev => ({

...prev,

[title]: !prev[title]

}));

};

const handleSubmit = (e: React.FormEvent) => {

e.preventDefault();

const featuresArray = Object.values(values);

onSubmit(featuresArray);

};

return (

<div className="w-full p-4 bg-white rounded-lg shadow-md mt-4">

<h2 className="text-xl font-bold mb-4 text-gray-800">Input UHPC Joint Feature Data</h2>

<form onSubmit={handleSubmit} className="space-y-6">

{fieldGroups.map(group => (

<div key={group.title} className="border rounded-lg overflow-hidden">

<button

type="button"

className="w-full px-4 py-3 bg-gray-100 text-left font-semibold text-gray-800 hover:bg-gray-200 flex justify-between items-center"

onClick={() => toggleGroup(group.title)}

>

<span>{group.title}</span>

<span>{expanded[group.title] ? '▼' : '►'}</span>

</button>

{expanded[group.title] && (

<div className="p-4 grid grid-cols-1 md:grid-cols-2 gap-4">

{group.fields.map(field => (

<div key={field.id} className="space-y-1">

<label className="block text-gray-800 font-medium">

{field.label}

</label>

<div className="text-xs text-gray-600 mb-1">{field.description}</div>

{field.type === 'select' ? (

<select

value={values[field.id]}

onChange={(e) => handleChange(field.id, Number(e.target.value))}

className="w-full p-2 border rounded focus:outline-none focus:ring-2 focus:ring-blue-500 text-gray-800"

>

{field.options && field.options.map((option: {value: number, label: string}) => (

<option key={option.value} value={option.value}>

{option.label}

</option>

))}

</select>

) : (

<input

type="number"

step={field.step || 1}

min={field.min !== undefined ? field.min : undefined}

value={values[field.id]}

onChange={(e) => handleChange(field.id, Number(e.target.value))}

className="w-full p-2 border rounded focus:outline-none focus:ring-2 focus:ring-blue-500 text-gray-800"

/>

)}

</div>

))}

</div>

)}

</div>

))}

<button

type="submit"

disabled={isLoading}

className={`w-full py-3 px-4 rounded font-semibold ${

isLoading

? 'bg-gray-400 cursor-not-allowed'

: 'bg-blue-600 hover:bg-blue-700 text-white'

}`}

>

{isLoading ? 'Predicting...' : 'Start Prediction'}

</button>

</form>

</div>

);

};

export default FeatureInput;

'use client';

import React from 'react';

type Model = {

name: string;

accuracy: number;

predictTime: string;

};

type ModelSelectorProps = {

models: Model[];

selectedModel: string;

onSelectModel: (model: string) => void;

};

const ModelSelector: React.FC<ModelSelectorProps> = ({

models,

selectedModel,

onSelectModel

}) => {

return (

<div className="w-full p-4 bg-white rounded-lg shadow-md">

<h2 className="text-xl font-bold mb-4 text-gray-800">选择预测模型</h2>

<div className="grid grid-cols-1 md:grid-cols-2 gap-4">

{models.map((model, index) => (

<div

key={index}

className={`p-4 border rounded-lg cursor-pointer transition-all duration-300 ${

selectedModel === (index === 0 ? 'randomForest' : 'optimalNN')

? 'bg-blue-50 border-blue-500 shadow-md'

: 'border-gray-200 hover:border-blue-300'

}`}

onClick={() => onSelectModel(index === 0 ? 'randomForest' : 'optimalNN')}

>

<h3 className="font-semibold text-lg text-gray-800">{model.name}</h3>

<div className="mt-2 text-sm text-gray-600">

<p>准确率: <span className="font-medium">{(model.accuracy \* 100).toFixed(1)}%</span></p>

<p>平均预测时间: <span className="font-medium">{model.predictTime}ms</span></p>

</div>

</div>

))}

</div>

</div>

);

};

export default ModelSelector;

'use client';

import React from 'react';

type PredictionResultProps = {

result: {

modelName: string;

confidence: number;

processTime: string;

individual\_predictions: number[];

shear\_capacity: number;

} | null;

};

const PredictionResult: React.FC<PredictionResultProps> = ({ result }) => {

if (!result) return null;

const { modelName, confidence, processTime, individual\_predictions, shear\_capacity } = result;

return (

<div className="w-full p-6 bg-white rounded-lg shadow-md mt-6 border-t-4 border-blue-500">

<h2 className="text-2xl font-bold mb-6 text-gray-800 flex items-center">

<svg xmlns="http://www.w3.org/2000/svg" className="h-6 w-6 mr-2 text-blue-500" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M9 5H7a2 2 0 00-2 2v12a2 2 0 002 2h10a2 2 0 002-2V7a2 2 0 00-2-2h-2M9 5a2 2 0 002 2h2a2 2 0 002-2M9 5a2 2 0 012-2h2a2 2 0 012 2" />

</svg>

预测结果

</h2>

<div className="grid md:grid-cols-2 gap-6">

<div className="bg-blue-50 p-4 rounded-lg">

<div className="flex items-center mb-4">

<div className="rounded-full bg-blue-100 p-2 mr-3">

<svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 text-blue-600" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M13 10V3L4 14h7v7l9-11h-7z" />

</svg>

</div>

<h3 className="text-lg font-semibold text-gray-800">模型信息</h3>

</div>

<div className="space-y-3">

<div className="flex items-center justify-between py-2 border-b border-blue-100">

<span className="text-gray-700">使用模型:</span>

<span className="font-semibold text-blue-800">{modelName}</span>

</div>

<div className="flex items-center justify-between py-2 border-b border-blue-100">

<span className="text-gray-700">置信度:</span>

<div className="flex items-center">

<div className="w-32 bg-gray-200 rounded-full h-2.5 mr-2">

<div

className="bg-blue-600 h-2.5 rounded-full"

style={{ width: `${confidence \* 100}%` }}

></div>

</div>

<span className="font-semibold text-gray-700">{(confidence \* 100).toFixed(1)}%</span>

</div>

</div>

<div className="flex items-center justify-between py-2">

<span className="text-gray-700">处理时间:</span>

<span className="font-semibold text-gray-700">{processTime}</span>

</div>

</div>

</div>

<div className="bg-green-50 p-4 rounded-lg">

<div className="flex items-center mb-4">

<div className="rounded-full bg-green-100 p-2 mr-3">

<svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 text-green-600" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M9 19v-6a2 2 0 00-2-2H5a2 2 0 00-2 2v6a2 2 0 002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0 012 2v10m-6 0a2 2 0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0 012 2v14a2 2 0 01-2 2h-2a2 2 0 01-2-2z" />

</svg>

</div>

<h3 className="text-lg font-semibold text-gray-800">预测结果</h3>

</div>

<div className="space-y-3">

<div className="flex items-center justify-between py-2 border-b border-green-100">

<span className="text-gray-700">抗剪承载力:</span>

<span className="font-bold text-green-700">{shear\_capacity.toFixed(2)} kN</span>

</div>

</div>

</div>

</div>

<div className="mt-6 bg-gray-50 p-4 rounded-lg">

<div className="flex items-center mb-4">

<div className="rounded-full bg-gray-200 p-2 mr-3">

<svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 text-gray-600" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M9 17v-2m3 2v-4m3 4v-6m2 10H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z" />

</svg>

</div>

<h3 className="text-lg font-semibold text-gray-800">个别预测明细</h3>

</div>

<div className="overflow-x-auto">

<table className="min-w-full divide-y divide-gray-200">

<thead className="bg-gray-100">

<tr>

<th scope="col" className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

预测序号

</th>

<th scope="col" className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

预测值 (kN)

</th>

<th scope="col" className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

与平均值偏差

</th>

</tr>

</thead>

<tbody className="bg-white divide-y divide-gray-200">

{individual\_predictions.map((prediction, index) => {

const deviation = prediction - shear\_capacity;

const deviationPercent = (deviation / shear\_capacity) \* 100;

return (

<tr key={index}>

<td className="px-6 py-4 whitespace-nowrap text-sm font-medium text-gray-900">

预测 {index + 1}

</td>

<td className="px-6 py-4 whitespace-nowrap text-sm text-gray-500">

{prediction.toFixed(2)}

</td>

<td className="px-6 py-4 whitespace-nowrap text-sm">

<span className={`px-2 py-1 rounded-full text-xs font-semibold ${

deviationPercent > 0 ? 'bg-green-100 text-green-800' : 'bg-red-100 text-red-800'

}`}>

{deviationPercent > 0 ? '+' : ''}{deviationPercent.toFixed(2)}%

</span>

</td>

</tr>

);

})}

</tbody>

</table>

</div>

</div>

<div className="mt-6 p-4 bg-yellow-50 rounded-lg border border-yellow-100">

<div className="flex items-start">

<div className="flex-shrink-0">

<svg className="h-5 w-5 text-yellow-400" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20" fill="currentColor">

<path fillRule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clipRule="evenodd" />

</svg>

</div>

<div className="ml-3">

<h3 className="text-sm font-medium text-yellow-800">

模型说明

</h3>

<div className="mt-2 text-sm text-yellow-700">

<p>

该预测结果仅供参考，实际抗剪承载力可能受多种因素影响。最终结果是基于多个预测取平均值得出。

</p>

</div>

</div>

</div>

</div>

</div>

);

};

export default PredictionResult;

@import "tailwindcss";

:root {

--background: #ffffff;

--foreground: #171717;

}

@theme inline {

--color-background: var(--background);

--color-foreground: var(--foreground);

--font-sans: var(--font-geist-sans);

--font-mono: var(--font-geist-mono);

}

@media (prefers-color-scheme: dark) {

:root {

--background: #0a0a0a;

--foreground: #ededed;

}

}

body {

background: var(--background);

color: var(--foreground);

font-family: Arial, Helvetica, sans-serif;

}

import type { Metadata } from "next";

import { Inter } from "next/font/google";

import "./globals.css";

const inter = Inter({ subsets: ["latin"] });

export const metadata: Metadata = {

title: "多模型机器学习预测系统",

description: "基于Next.js和Express构建的多模型机器学习预测系统",

authors: [{ name: "李嘉俊", url: "https://github.com" }],

creator: "长沙理工大学 李嘉俊",

};

export default function RootLayout({

children,

}: Readonly<{

children: React.ReactNode;

}>) {

return (

<html lang="zh-CN">

<body className={inter.className}>{children}</body>

</html>

);

}

'use client';

import { useState, useEffect } from 'react';

import ModelSelector from './components/ModelSelector';

import FeatureInput from './components/FeatureInput';

import PredictionResult from './components/PredictionResult';

const API\_URL = 'http://localhost:5000';

type Model = {

name: string;

accuracy: number;

predictTime: string;

};

type PredictionResult = {

modelName: string;

confidence: number;

processTime: string;

individual\_predictions: number[];

shear\_capacity: number;

};

export default function Home() {

const [models, setModels] = useState<Model[]>([]);

const [selectedModel, setSelectedModel] = useState<string>('randomForest');

const [loading, setLoading] = useState<boolean>(false);

const [result, setResult] = useState<PredictionResult | null>(null);

const [error, setError] = useState<string>('');

useEffect(() => {

async function fetchModels() {

try {

const response = await fetch(`${API\_URL}/api/models`);

if (!response.ok) {

throw new Error('获取模型列表失败');

}

const data = await response.json();

setModels(data);

} catch (err) {

console.error('获取模型时出错:', err);

setError('无法连接到服务器。请确保后端服务正在运行。');

}

}

fetchModels();

}, []);

const handleSelectModel = (model: string) => {

setSelectedModel(model);

};

const handleSubmitFeatures = async (features: number[]) => {

setLoading(true);

setError('');

try {

const response = await fetch(`${API\_URL}/api/predict`, {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({

modelType: selectedModel,

features: features,

}),

});

if (!response.ok) {

throw new Error('预测请求失败');

}

const data = await response.json();

setResult(data);

} catch (err) {

console.error('预测过程中出错:', err);

setError('预测失败。请稍后再试。');

} finally {

setLoading(false);

}

};

const handleDemoResult = () => {

setResult({

modelName: "随机森林",

confidence: 0.8008928720291373,

processTime: "0ms",

individual\_predictions: [383.7, 595.7, 620.1],

shear\_capacity: 533.1666666666666

});

};

return (

<main className="min-h-screen p-4 md:p-8 bg-gray-50">

<div className="max-w-5xl mx-auto">

<header className="text-center mb-8">

<h1 className="text-3xl md:text-4xl font-bold text-gray-800 mb-2">

多模型机器学习预测系统

</h1>

<p className="text-gray-600">

选择模型、输入特征数据，获取智能预测结果

</p>

</header>

{error && (

<div className="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded mb-6">

<p>{error}</p>

</div>

)}

{models.length > 0 ? (

<>

<ModelSelector

models={models}

selectedModel={selectedModel}

onSelectModel={handleSelectModel}

/>

<FeatureInput

onSubmit={handleSubmitFeatures}

isLoading={loading}

/>

{result && <PredictionResult result={result} />}

{/\* 开发环境使用，方便测试UI \*/}

{process.env.NODE\_ENV === 'development' && !result && (

<div className="mt-4">

<button

onClick={handleDemoResult}

className="text-sm text-gray-500 underline"

>

显示示例结果（仅开发环境）

</button>

</div>

)}

</>

) : !error ? (

<div className="text-center py-8">

<div className="inline-block h-8 w-8 animate-spin rounded-full border-4 border-solid border-blue-600 border-r-transparent"></div>

<p className="mt-4 text-gray-600">加载模型中...</p>

</div>

) : null}

</div>

<footer className="text-center mt-12 py-4 text-sm text-gray-500 border-t">

<p>© {new Date().getFullYear()} 长沙理工大学 - 李嘉俊</p>

</footer>

</main>

);

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta http-equiv="X-UA-Compatible" content="ie=edge" />

<title>eventsource-parser@3.0.2 bundle analysis</title>

<style>

:root {

--font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial,

"Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol",

"Noto Color Emoji";

--background-color: #2b2d42;

--text-color: #edf2f4;

}

html {

box-sizing: border-box;

}

\*,

\*:before,

\*:after {

box-sizing: inherit;

}

html {

background-color: var(--background-color);

color: var(--text-color);

font-family: var(--font-family);

}

body {

padding: 0;

margin: 0;

}

html,

body {

height: 100%;

width: 100%;

overflow: hidden;

}

body {

display: flex;

flex-direction: column;

}

svg {

vertical-align: middle;

width: 100%;

height: 100%;

max-height: 100vh;

}

main {

flex-grow: 1;

height: 100vh;

padding: 20px;

}

.tooltip {

position: absolute;

z-index: 1070;

border: 2px solid;

border-radius: 5px;

padding: 5px;

font-size: 0.875rem;

background-color: var(--background-color);

color: var(--text-color);

}

.tooltip-hidden {

visibility: hidden;

opacity: 0;

}

.sidebar {

position: fixed;

top: 0;

left: 0;

right: 0;

display: flex;

flex-direction: row;

font-size: 0.7rem;

align-items: center;

margin: 0 50px;

height: 20px;

}

.size-selectors {

display: flex;

flex-direction: row;

align-items: center;

}

.size-selector {

display: flex;

flex-direction: row;

align-items: center;

justify-content: center;

margin-right: 1rem;

}

.size-selector input {

margin: 0 0.3rem 0 0;

}

.filters {

flex: 1;

display: flex;

flex-direction: row;

align-items: center;

}

.module-filters {

display: flex;

flex-grow: 1;

}

.module-filter {

display: flex;

flex-direction: row;

align-items: center;

justify-content: center;

flex: 1;

}

.module-filter input {

flex: 1;

height: 1rem;

padding: 0.01rem;

font-size: 0.7rem;

margin-left: 0.3rem;

}

.module-filter + .module-filter {

margin-left: 0.5rem;

}

.node {

cursor: pointer;

}

</style>

</head>

<body>

<main></main>

<script>

/\*<!--\*/

var drawChart = (function (exports) {

'use strict';

var n,l$1,u$2,i$1,r$1,o$1,e$1,f$2,c$1,s$1,a$1,h$1,p$1={},v$1=[],y$1=/acit|ex(?:s|g|n|p|$)|rph|grid|ows|mnc|ntw|ine[ch]|zoo|^ord|itera/i,d$1=Array.isArray;function w$1(n,l){for(var u in l)n[u]=l[u];return n}function \_$1(n){n&&n.parentNode&&n.parentNode.removeChild(n);}function g(l,u,t){var i,r,o,e={};for(o in u)"key"==o?i=u[o]:"ref"==o?r=u[o]:e[o]=u[o];if(arguments.length>2&&(e.children=arguments.length>3?n.call(arguments,2):t),"function"==typeof l&&null!=l.defaultProps)for(o in l.defaultProps)void 0===e[o]&&(e[o]=l.defaultProps[o]);return m$1(l,e,i,r,null)}function m$1(n,t,i,r,o){var e={type:n,props:t,key:i,ref:r,\_\_k:null,\_\_:null,\_\_b:0,\_\_e:null,\_\_c:null,constructor:void 0,\_\_v:null==o?++u$2:o,\_\_i:-1,\_\_u:0};return null==o&&null!=l$1.vnode&&l$1.vnode(e),e}function k$1(n){return n.children}function x$1(n,l){this.props=n,this.context=l;}function C$1(n,l){if(null==l)return n.\_\_?C$1(n.\_\_,n.\_\_i+1):null;for(var u;l<n.\_\_k.length;l++)if(null!=(u=n.\_\_k[l])&&null!=u.\_\_e)return u.\_\_e;return "function"==typeof n.type?C$1(n):null}function S(n){var l,u;if(null!=(n=n.\_\_)&&null!=n.\_\_c){for(n.\_\_e=n.\_\_c.base=null,l=0;l<n.\_\_k.length;l++)if(null!=(u=n.\_\_k[l])&&null!=u.\_\_e){n.\_\_e=n.\_\_c.base=u.\_\_e;break}return S(n)}}function M(n){(!n.\_\_d&&(n.\_\_d=!0)&&i$1.push(n)&&!P.\_\_r++||r$1!==l$1.debounceRendering)&&((r$1=l$1.debounceRendering)||o$1)(P);}function P(){var n,u,t,r,o,f,c,s;for(i$1.sort(e$1);n=i$1.shift();)n.\_\_d&&(u=i$1.length,r=void 0,f=(o=(t=n).\_\_v).\_\_e,c=[],s=[],t.\_\_P&&((r=w$1({},o)).\_\_v=o.\_\_v+1,l$1.vnode&&l$1.vnode(r),j$1(t.\_\_P,r,o,t.\_\_n,t.\_\_P.namespaceURI,32&o.\_\_u?[f]:null,c,null==f?C$1(o):f,!!(32&o.\_\_u),s),r.\_\_v=o.\_\_v,r.\_\_.\_\_k[r.\_\_i]=r,z$1(c,r,s),r.\_\_e!=f&&S(r)),i$1.length>u&&i$1.sort(e$1));P.\_\_r=0;}function $(n,l,u,t,i,r,o,e,f,c,s){var a,h,y,d,w,\_,g=t&&t.\_\_k||v$1,m=l.length;for(f=I(u,l,g,f,m),a=0;a<m;a++)null!=(y=u.\_\_k[a])&&(h=-1===y.\_\_i?p$1:g[y.\_\_i]||p$1,y.\_\_i=a,\_=j$1(n,y,h,i,r,o,e,f,c,s),d=y.\_\_e,y.ref&&h.ref!=y.ref&&(h.ref&&V(h.ref,null,y),s.push(y.ref,y.\_\_c||d,y)),null==w&&null!=d&&(w=d),4&y.\_\_u||h.\_\_k===y.\_\_k?f=A$1(y,f,n):"function"==typeof y.type&&void 0!==\_?f=\_:d&&(f=d.nextSibling),y.\_\_u&=-7);return u.\_\_e=w,f}function I(n,l,u,t,i){var r,o,e,f,c,s=u.length,a=s,h=0;for(n.\_\_k=new Array(i),r=0;r<i;r++)null!=(o=l[r])&&"boolean"!=typeof o&&"function"!=typeof o?(f=r+h,(o=n.\_\_k[r]="string"==typeof o||"number"==typeof o||"bigint"==typeof o||o.constructor==String?m$1(null,o,null,null,null):d$1(o)?m$1(k$1,{children:o},null,null,null):void 0===o.constructor&&o.\_\_b>0?m$1(o.type,o.props,o.key,o.ref?o.ref:null,o.\_\_v):o).\_\_=n,o.\_\_b=n.\_\_b+1,e=null,-1!==(c=o.\_\_i=L(o,u,f,a))&&(a--,(e=u[c])&&(e.\_\_u|=2)),null==e||null===e.\_\_v?(-1==c&&h--,"function"!=typeof o.type&&(o.\_\_u|=4)):c!=f&&(c==f-1?h--:c==f+1?h++:(c>f?h--:h++,o.\_\_u|=4))):n.\_\_k[r]=null;if(a)for(r=0;r<s;r++)null!=(e=u[r])&&0==(2&e.\_\_u)&&(e.\_\_e==t&&(t=C$1(e)),q$1(e,e));return t}function A$1(n,l,u){var t,i;if("function"==typeof n.type){for(t=n.\_\_k,i=0;t&&i<t.length;i++)t[i]&&(t[i].\_\_=n,l=A$1(t[i],l,u));return l}n.\_\_e!=l&&(l&&n.type&&!u.contains(l)&&(l=C$1(n)),u.insertBefore(n.\_\_e,l||null),l=n.\_\_e);do{l=l&&l.nextSibling;}while(null!=l&&8==l.nodeType);return l}function L(n,l,u,t){var i,r,o=n.key,e=n.type,f=l[u];if(null===f||f&&o==f.key&&e===f.type&&0==(2&f.\_\_u))return u;if(t>(null!=f&&0==(2&f.\_\_u)?1:0))for(i=u-1,r=u+1;i>=0||r<l.length;){if(i>=0){if((f=l[i])&&0==(2&f.\_\_u)&&o==f.key&&e===f.type)return i;i--;}if(r<l.length){if((f=l[r])&&0==(2&f.\_\_u)&&o==f.key&&e===f.type)return r;r++;}}return -1}function T$1(n,l,u){"-"==l[0]?n.setProperty(l,null==u?"":u):n[l]=null==u?"":"number"!=typeof u||y$1.test(l)?u:u+"px";}function F(n,l,u,t,i){var r;n:if("style"==l)if("string"==typeof u)n.style.cssText=u;else {if("string"==typeof t&&(n.style.cssText=t=""),t)for(l in t)u&&l in u||T$1(n.style,l,"");if(u)for(l in u)t&&u[l]===t[l]||T$1(n.style,l,u[l]);}else if("o"==l[0]&&"n"==l[1])r=l!=(l=l.replace(f$2,"$1")),l=l.toLowerCase()in n||"onFocusOut"==l||"onFocusIn"==l?l.toLowerCase().slice(2):l.slice(2),n.l||(n.l={}),n.l[l+r]=u,u?t?u.u=t.u:(u.u=c$1,n.addEventListener(l,r?a$1:s$1,r)):n.removeEventListener(l,r?a$1:s$1,r);else {if("http://www.w3.org/2000/svg"==i)l=l.replace(/xlink(H|:h)/,"h").replace(/sName$/,"s");else if("width"!=l&&"height"!=l&&"href"!=l&&"list"!=l&&"form"!=l&&"tabIndex"!=l&&"download"!=l&&"rowSpan"!=l&&"colSpan"!=l&&"role"!=l&&"popover"!=l&&l in n)try{n[l]=null==u?"":u;break n}catch(n){}"function"==typeof u||(null==u||!1===u&&"-"!=l[4]?n.removeAttribute(l):n.setAttribute(l,"popover"==l&&1==u?"":u));}}function O(n){return function(u){if(this.l){var t=this.l[u.type+n];if(null==u.t)u.t=c$1++;else if(u.t<t.u)return;return t(l$1.event?l$1.event(u):u)}}}function j$1(n,u,t,i,r,o,e,f,c,s){var a,h,p,v,y,g,m,b,C,S,M,P,I,A,H,L,T,F=u.type;if(void 0!==u.constructor)return null;128&t.\_\_u&&(c=!!(32&t.\_\_u),o=[f=u.\_\_e=t.\_\_e]),(a=l$1.\_\_b)&&a(u);n:if("function"==typeof F)try{if(b=u.props,C="prototype"in F&&F.prototype.render,S=(a=F.contextType)&&i[a.\_\_c],M=a?S?S.props.value:a.\_\_:i,t.\_\_c?m=(h=u.\_\_c=t.\_\_c).\_\_=h.\_\_E:(C?u.\_\_c=h=new F(b,M):(u.\_\_c=h=new x$1(b,M),h.constructor=F,h.render=B$1),S&&S.sub(h),h.props=b,h.state||(h.state={}),h.context=M,h.\_\_n=i,p=h.\_\_d=!0,h.\_\_h=[],h.\_sb=[]),C&&null==h.\_\_s&&(h.\_\_s=h.state),C&&null!=F.getDerivedStateFromProps&&(h.\_\_s==h.state&&(h.\_\_s=w$1({},h.\_\_s)),w$1(h.\_\_s,F.getDerivedStateFromProps(b,h.\_\_s))),v=h.props,y=h.state,h.\_\_v=u,p)C&&null==F.getDerivedStateFromProps&&null!=h.componentWillMount&&h.componentWillMount(),C&&null!=h.componentDidMount&&h.\_\_h.push(h.componentDidMount);else {if(C&&null==F.getDerivedStateFromProps&&b!==v&&null!=h.componentWillReceiveProps&&h.componentWillReceiveProps(b,M),!h.\_\_e&&(null!=h.shouldComponentUpdate&&!1===h.shouldComponentUpdate(b,h.\_\_s,M)||u.\_\_v==t.\_\_v)){for(u.\_\_v!=t.\_\_v&&(h.props=b,h.state=h.\_\_s,h.\_\_d=!1),u.\_\_e=t.\_\_e,u.\_\_k=t.\_\_k,u.\_\_k.some(function(n){n&&(n.\_\_=u);}),P=0;P<h.\_sb.length;P++)h.\_\_h.push(h.\_sb[P]);h.\_sb=[],h.\_\_h.length&&e.push(h);break n}null!=h.componentWillUpdate&&h.componentWillUpdate(b,h.\_\_s,M),C&&null!=h.componentDidUpdate&&h.\_\_h.push(function(){h.componentDidUpdate(v,y,g);});}if(h.context=M,h.props=b,h.\_\_P=n,h.\_\_e=!1,I=l$1.\_\_r,A=0,C){for(h.state=h.\_\_s,h.\_\_d=!1,I&&I(u),a=h.render(h.props,h.state,h.context),H=0;H<h.\_sb.length;H++)h.\_\_h.push(h.\_sb[H]);h.\_sb=[];}else do{h.\_\_d=!1,I&&I(u),a=h.render(h.props,h.state,h.context),h.state=h.\_\_s;}while(h.\_\_d&&++A<25);h.state=h.\_\_s,null!=h.getChildContext&&(i=w$1(w$1({},i),h.getChildContext())),C&&!p&&null!=h.getSnapshotBeforeUpdate&&(g=h.getSnapshotBeforeUpdate(v,y)),f=$(n,d$1(L=null!=a&&a.type===k$1&&null==a.key?a.props.children:a)?L:[L],u,t,i,r,o,e,f,c,s),h.base=u.\_\_e,u.\_\_u&=-161,h.\_\_h.length&&e.push(h),m&&(h.\_\_E=h.\_\_=null);}catch(n){if(u.\_\_v=null,c||null!=o)if(n.then){for(u.\_\_u|=c?160:128;f&&8==f.nodeType&&f.nextSibling;)f=f.nextSibling;o[o.indexOf(f)]=null,u.\_\_e=f;}else for(T=o.length;T--;)\_$1(o[T]);else u.\_\_e=t.\_\_e,u.\_\_k=t.\_\_k;l$1.\_\_e(n,u,t);}else null==o&&u.\_\_v==t.\_\_v?(u.\_\_k=t.\_\_k,u.\_\_e=t.\_\_e):f=u.\_\_e=N(t.\_\_e,u,t,i,r,o,e,c,s);return (a=l$1.diffed)&&a(u),128&u.\_\_u?void 0:f}function z$1(n,u,t){for(var i=0;i<t.length;i++)V(t[i],t[++i],t[++i]);l$1.\_\_c&&l$1.\_\_c(u,n),n.some(function(u){try{n=u.\_\_h,u.\_\_h=[],n.some(function(n){n.call(u);});}catch(n){l$1.\_\_e(n,u.\_\_v);}});}function N(u,t,i,r,o,e,f,c,s){var a,h,v,y,w,g,m,b=i.props,k=t.props,x=t.type;if("svg"==x?o="http://www.w3.org/2000/svg":"math"==x?o="http://www.w3.org/1998/Math/MathML":o||(o="http://www.w3.org/1999/xhtml"),null!=e)for(a=0;a<e.length;a++)if((w=e[a])&&"setAttribute"in w==!!x&&(x?w.localName==x:3==w.nodeType)){u=w,e[a]=null;break}if(null==u){if(null==x)return document.createTextNode(k);u=document.createElementNS(o,x,k.is&&k),c&&(l$1.\_\_m&&l$1.\_\_m(t,e),c=!1),e=null;}if(null===x)b===k||c&&u.data===k||(u.data=k);else {if(e=e&&n.call(u.childNodes),b=i.props||p$1,!c&&null!=e)for(b={},a=0;a<u.attributes.length;a++)b[(w=u.attributes[a]).name]=w.value;for(a in b)if(w=b[a],"children"==a);else if("dangerouslySetInnerHTML"==a)v=w;else if(!(a in k)){if("value"==a&&"defaultValue"in k||"checked"==a&&"defaultChecked"in k)continue;F(u,a,null,w,o);}for(a in k)w=k[a],"children"==a?y=w:"dangerouslySetInnerHTML"==a?h=w:"value"==a?g=w:"checked"==a?m=w:c&&"function"!=typeof w||b[a]===w||F(u,a,w,b[a],o);if(h)c||v&&(h.\_\_html===v.\_\_html||h.\_\_html===u.innerHTML)||(u.innerHTML=h.\_\_html),t.\_\_k=[];else if(v&&(u.innerHTML=""),$(u,d$1(y)?y:[y],t,i,r,"foreignObject"==x?"http://www.w3.org/1999/xhtml":o,e,f,e?e[0]:i.\_\_k&&C$1(i,0),c,s),null!=e)for(a=e.length;a--;)\_$1(e[a]);c||(a="value","progress"==x&&null==g?u.removeAttribute("value"):void 0!==g&&(g!==u[a]||"progress"==x&&!g||"option"==x&&g!==b[a])&&F(u,a,g,b[a],o),a="checked",void 0!==m&&m!==u[a]&&F(u,a,m,b[a],o));}return u}function V(n,u,t){try{if("function"==typeof n){var i="function"==typeof n.\_\_u;i&&n.\_\_u(),i&&null==u||(n.\_\_u=n(u));}else n.current=u;}catch(n){l$1.\_\_e(n,t);}}function q$1(n,u,t){var i,r;if(l$1.unmount&&l$1.unmount(n),(i=n.ref)&&(i.current&&i.current!==n.\_\_e||V(i,null,u)),null!=(i=n.\_\_c)){if(i.componentWillUnmount)try{i.componentWillUnmount();}catch(n){l$1.\_\_e(n,u);}i.base=i.\_\_P=null;}if(i=n.\_\_k)for(r=0;r<i.length;r++)i[r]&&q$1(i[r],u,t||"function"!=typeof n.type);t||\_$1(n.\_\_e),n.\_\_c=n.\_\_=n.\_\_e=void 0;}function B$1(n,l,u){return this.constructor(n,u)}function D$1(u,t,i){var r,o,e,f;t==document&&(t=document.documentElement),l$1.\_\_&&l$1.\_\_(u,t),o=(r="function"==typeof i)?null:t.\_\_k,e=[],f=[],j$1(t,u=(t).\_\_k=g(k$1,null,[u]),o||p$1,p$1,t.namespaceURI,o?null:t.firstChild?n.call(t.childNodes):null,e,o?o.\_\_e:t.firstChild,r,f),z$1(e,u,f);}function J(n,l){var u={\_\_c:l="\_\_cC"+h$1++,\_\_:n,Consumer:function(n,l){return n.children(l)},Provider:function(n){var u,t;return this.getChildContext||(u=new Set,(t={})[l]=this,this.getChildContext=function(){return t},this.componentWillUnmount=function(){u=null;},this.shouldComponentUpdate=function(n){this.props.value!==n.value&&u.forEach(function(n){n.\_\_e=!0,M(n);});},this.sub=function(n){u.add(n);var l=n.componentWillUnmount;n.componentWillUnmount=function(){u&&u.delete(n),l&&l.call(n);};}),n.children}};return u.Provider.\_\_=u.Consumer.contextType=u}n=v$1.slice,l$1={\_\_e:function(n,l,u,t){for(var i,r,o;l=l.\_\_;)if((i=l.\_\_c)&&!i.\_\_)try{if((r=i.constructor)&&null!=r.getDerivedStateFromError&&(i.setState(r.getDerivedStateFromError(n)),o=i.\_\_d),null!=i.componentDidCatch&&(i.componentDidCatch(n,t||{}),o=i.\_\_d),o)return i.\_\_E=i}catch(l){n=l;}throw n}},u$2=0,x$1.prototype.setState=function(n,l){var u;u=null!=this.\_\_s&&this.\_\_s!==this.state?this.\_\_s:this.\_\_s=w$1({},this.state),"function"==typeof n&&(n=n(w$1({},u),this.props)),n&&w$1(u,n),null!=n&&this.\_\_v&&(l&&this.\_sb.push(l),M(this));},x$1.prototype.forceUpdate=function(n){this.\_\_v&&(this.\_\_e=!0,n&&this.\_\_h.push(n),M(this));},x$1.prototype.render=k$1,i$1=[],o$1="function"==typeof Promise?Promise.prototype.then.bind(Promise.resolve()):setTimeout,e$1=function(n,l){return n.\_\_v.\_\_b-l.\_\_v.\_\_b},P.\_\_r=0,f$2=/(PointerCapture)$|Capture$/i,c$1=0,s$1=O(!1),a$1=O(!0),h$1=0;

var f$1=0;function u$1(e,t,n,o,i,u){t||(t={});var a,c,p=t;if("ref"in p)for(c in p={},t)"ref"==c?a=t[c]:p[c]=t[c];var l={type:e,props:p,key:n,ref:a,\_\_k:null,\_\_:null,\_\_b:0,\_\_e:null,\_\_c:null,constructor:void 0,\_\_v:--f$1,\_\_i:-1,\_\_u:0,\_\_source:i,\_\_self:u};if("function"==typeof e&&(a=e.defaultProps))for(c in a)void 0===p[c]&&(p[c]=a[c]);return l$1.vnode&&l$1.vnode(l),l}

function count$1(node) {

var sum = 0,

children = node.children,

i = children && children.length;

if (!i) sum = 1;

else while (--i >= 0) sum += children[i].value;

node.value = sum;

}

function node\_count() {

return this.eachAfter(count$1);

}

function node\_each(callback, that) {

let index = -1;

for (const node of this) {

callback.call(that, node, ++index, this);

}

return this;

}

function node\_eachBefore(callback, that) {

var node = this, nodes = [node], children, i, index = -1;

while (node = nodes.pop()) {

callback.call(that, node, ++index, this);

if (children = node.children) {

for (i = children.length - 1; i >= 0; --i) {

nodes.push(children[i]);

}

}

}

return this;

}

function node\_eachAfter(callback, that) {

var node = this, nodes = [node], next = [], children, i, n, index = -1;

while (node = nodes.pop()) {

next.push(node);

if (children = node.children) {

for (i = 0, n = children.length; i < n; ++i) {

nodes.push(children[i]);

}

}

}

while (node = next.pop()) {

callback.call(that, node, ++index, this);

}

return this;

}

function node\_find(callback, that) {

let index = -1;

for (const node of this) {

if (callback.call(that, node, ++index, this)) {

return node;

}

}

}

function node\_sum(value) {

return this.eachAfter(function(node) {

var sum = +value(node.data) || 0,

children = node.children,

i = children && children.length;

while (--i >= 0) sum += children[i].value;

node.value = sum;

});

}

function node\_sort(compare) {

return this.eachBefore(function(node) {

if (node.children) {

node.children.sort(compare);

}

});

}

function node\_path(end) {

var start = this,

ancestor = leastCommonAncestor(start, end),

nodes = [start];

while (start !== ancestor) {

start = start.parent;

nodes.push(start);

}

var k = nodes.length;

while (end !== ancestor) {

nodes.splice(k, 0, end);

end = end.parent;

}

return nodes;

}

function leastCommonAncestor(a, b) {

if (a === b) return a;

var aNodes = a.ancestors(),

bNodes = b.ancestors(),

c = null;

a = aNodes.pop();

b = bNodes.pop();

while (a === b) {

c = a;

a = aNodes.pop();

b = bNodes.pop();

}

return c;

}

function node\_ancestors() {

var node = this, nodes = [node];

while (node = node.parent) {

nodes.push(node);

}

return nodes;

}

function node\_descendants() {

return Array.from(this);

}

function node\_leaves() {

var leaves = [];

this.eachBefore(function(node) {

if (!node.children) {

leaves.push(node);

}

});

return leaves;

}

function node\_links() {

var root = this, links = [];

root.each(function(node) {

if (node !== root) { // Don’t include the root’s parent, if any.

links.push({source: node.parent, target: node});

}

});

return links;

}

function\* node\_iterator() {

var node = this, current, next = [node], children, i, n;

do {

current = next.reverse(), next = [];

while (node = current.pop()) {

yield node;

if (children = node.children) {

for (i = 0, n = children.length; i < n; ++i) {

next.push(children[i]);

}

}

}

} while (next.length);

}

function hierarchy(data, children) {

if (data instanceof Map) {

data = [undefined, data];

if (children === undefined) children = mapChildren;

} else if (children === undefined) {

children = objectChildren;

}

var root = new Node$1(data),

node,

nodes = [root],

child,

childs,

i,

n;

while (node = nodes.pop()) {

if ((childs = children(node.data)) && (n = (childs = Array.from(childs)).length)) {

node.children = childs;

for (i = n - 1; i >= 0; --i) {

nodes.push(child = childs[i] = new Node$1(childs[i]));

child.parent = node;

child.depth = node.depth + 1;

}

}

}

return root.eachBefore(computeHeight);

}

function node\_copy() {

return hierarchy(this).eachBefore(copyData);

}

function objectChildren(d) {

return d.children;

}

function mapChildren(d) {

return Array.isArray(d) ? d[1] : null;

}

function copyData(node) {

if (node.data.value !== undefined) node.value = node.data.value;

node.data = node.data.data;

}

function computeHeight(node) {

var height = 0;

do node.height = height;

while ((node = node.parent) && (node.height < ++height));

}

function Node$1(data) {

this.data = data;

this.depth =

this.height = 0;

this.parent = null;

}

Node$1.prototype = hierarchy.prototype = {

constructor: Node$1,

count: node\_count,

each: node\_each,

eachAfter: node\_eachAfter,

eachBefore: node\_eachBefore,

find: node\_find,

sum: node\_sum,

sort: node\_sort,

path: node\_path,

ancestors: node\_ancestors,

descendants: node\_descendants,

leaves: node\_leaves,

links: node\_links,

copy: node\_copy,

[Symbol.iterator]: node\_iterator

};

function required(f) {

if (typeof f !== "function") throw new Error;

return f;

}

function constantZero() {

return 0;

}

function constant$1(x) {

return function() {

return x;

};

}

function roundNode(node) {

node.x0 = Math.round(node.x0);

node.y0 = Math.round(node.y0);

node.x1 = Math.round(node.x1);

node.y1 = Math.round(node.y1);

}

function treemapDice(parent, x0, y0, x1, y1) {

var nodes = parent.children,

node,

i = -1,

n = nodes.length,

k = parent.value && (x1 - x0) / parent.value;

while (++i < n) {

node = nodes[i], node.y0 = y0, node.y1 = y1;

node.x0 = x0, node.x1 = x0 += node.value \* k;

}

}

function treemapSlice(parent, x0, y0, x1, y1) {

var nodes = parent.children,

node,

i = -1,

n = nodes.length,

k = parent.value && (y1 - y0) / parent.value;

while (++i < n) {

node = nodes[i], node.x0 = x0, node.x1 = x1;

node.y0 = y0, node.y1 = y0 += node.value \* k;

}

}

var phi = (1 + Math.sqrt(5)) / 2;

function squarifyRatio(ratio, parent, x0, y0, x1, y1) {

var rows = [],

nodes = parent.children,

row,

nodeValue,

i0 = 0,

i1 = 0,

n = nodes.length,

dx, dy,

value = parent.value,

sumValue,

minValue,

maxValue,

newRatio,

minRatio,

alpha,

beta;

while (i0 < n) {

dx = x1 - x0, dy = y1 - y0;

do sumValue = nodes[i1++].value; while (!sumValue && i1 < n);

minValue = maxValue = sumValue;

alpha = Math.max(dy / dx, dx / dy) / (value \* ratio);

beta = sumValue \* sumValue \* alpha;

minRatio = Math.max(maxValue / beta, beta / minValue);

for (; i1 < n; ++i1) {

sumValue += nodeValue = nodes[i1].value;

if (nodeValue < minValue) minValue = nodeValue;

if (nodeValue > maxValue) maxValue = nodeValue;

beta = sumValue \* sumValue \* alpha;

newRatio = Math.max(maxValue / beta, beta / minValue);

if (newRatio > minRatio) { sumValue -= nodeValue; break; }

minRatio = newRatio;

}

rows.push(row = {value: sumValue, dice: dx < dy, children: nodes.slice(i0, i1)});

if (row.dice) treemapDice(row, x0, y0, x1, value ? y0 += dy \* sumValue / value : y1);

else treemapSlice(row, x0, y0, value ? x0 += dx \* sumValue / value : x1, y1);

value -= sumValue, i0 = i1;

}

return rows;

}

var squarify = (function custom(ratio) {

function squarify(parent, x0, y0, x1, y1) {

squarifyRatio(ratio, parent, x0, y0, x1, y1);

}

squarify.ratio = function(x) {

return custom((x = +x) > 1 ? x : 1);

};

return squarify;

})(phi);

function treemap() {

var tile = squarify,

round = false,

dx = 1,

dy = 1,

paddingStack = [0],

paddingInner = constantZero,

paddingTop = constantZero,

paddingRight = constantZero,

paddingBottom = constantZero,

paddingLeft = constantZero;

function treemap(root) {

root.x0 =

root.y0 = 0;

root.x1 = dx;

root.y1 = dy;

root.eachBefore(positionNode);

paddingStack = [0];

if (round) root.eachBefore(roundNode);

return root;

}

function positionNode(node) {

var p = paddingStack[node.depth],

x0 = node.x0 + p,

y0 = node.y0 + p,

x1 = node.x1 - p,

y1 = node.y1 - p;

if (x1 < x0) x0 = x1 = (x0 + x1) / 2;

if (y1 < y0) y0 = y1 = (y0 + y1) / 2;

node.x0 = x0;

node.y0 = y0;

node.x1 = x1;

node.y1 = y1;

if (node.children) {

p = paddingStack[node.depth + 1] = paddingInner(node) / 2;

x0 += paddingLeft(node) - p;

y0 += paddingTop(node) - p;

x1 -= paddingRight(node) - p;

y1 -= paddingBottom(node) - p;

if (x1 < x0) x0 = x1 = (x0 + x1) / 2;

if (y1 < y0) y0 = y1 = (y0 + y1) / 2;

tile(node, x0, y0, x1, y1);

}

}

treemap.round = function(x) {

return arguments.length ? (round = !!x, treemap) : round;

};

treemap.size = function(x) {

return arguments.length ? (dx = +x[0], dy = +x[1], treemap) : [dx, dy];

};

treemap.tile = function(x) {

return arguments.length ? (tile = required(x), treemap) : tile;

};

treemap.padding = function(x) {

return arguments.length ? treemap.paddingInner(x).paddingOuter(x) : treemap.paddingInner();

};

treemap.paddingInner = function(x) {

return arguments.length ? (paddingInner = typeof x === "function" ? x : constant$1(+x), treemap) : paddingInner;

};

treemap.paddingOuter = function(x) {

return arguments.length ? treemap.paddingTop(x).paddingRight(x).paddingBottom(x).paddingLeft(x) : treemap.paddingTop();

};

treemap.paddingTop = function(x) {

return arguments.length ? (paddingTop = typeof x === "function" ? x : constant$1(+x), treemap) : paddingTop;

};

treemap.paddingRight = function(x) {

return arguments.length ? (paddingRight = typeof x === "function" ? x : constant$1(+x), treemap) : paddingRight;

};

treemap.paddingBottom = function(x) {

return arguments.length ? (paddingBottom = typeof x === "function" ? x : constant$1(+x), treemap) : paddingBottom;

};

treemap.paddingLeft = function(x) {

return arguments.length ? (paddingLeft = typeof x === "function" ? x : constant$1(+x), treemap) : paddingLeft;

};

return treemap;

}

var treemapResquarify = (function custom(ratio) {

function resquarify(parent, x0, y0, x1, y1) {

if ((rows = parent.\_squarify) && (rows.ratio === ratio)) {

var rows,

row,

nodes,

i,

j = -1,

n,

m = rows.length,

value = parent.value;

while (++j < m) {

row = rows[j], nodes = row.children;

for (i = row.value = 0, n = nodes.length; i < n; ++i) row.value += nodes[i].value;

if (row.dice) treemapDice(row, x0, y0, x1, value ? y0 += (y1 - y0) \* row.value / value : y1);

else treemapSlice(row, x0, y0, value ? x0 += (x1 - x0) \* row.value / value : x1, y1);

value -= row.value;

}

} else {

parent.\_squarify = rows = squarifyRatio(ratio, parent, x0, y0, x1, y1);

rows.ratio = ratio;

}

}

resquarify.ratio = function(x) {

return custom((x = +x) > 1 ? x : 1);

};

return resquarify;

})(phi);

const isModuleTree = (mod) => "children" in mod;

let count = 0;

class Id {

constructor(id) {

this.\_id = id;

const url = new URL(window.location.href);

url.hash = id;

this.\_href = url.toString();

}

get id() {

return this.\_id;

}

get href() {

return this.\_href;

}

toString() {

return `url(${this.href})`;

}

}

function generateUniqueId(name) {

count += 1;

const id = ["O", name, count].filter(Boolean).join("-");

return new Id(id);

}

const LABELS = {

renderedLength: "Rendered",

gzipLength: "Gzip",

brotliLength: "Brotli",

};

const getAvailableSizeOptions = (options) => {

const availableSizeProperties = ["renderedLength"];

if (options.gzip) {

availableSizeProperties.push("gzipLength");

}

if (options.brotli) {

availableSizeProperties.push("brotliLength");

}

return availableSizeProperties;

};

var t,r,u,i,o=0,f=[],c=l$1,e=c.\_\_b,a=c.\_\_r,v=c.diffed,l=c.\_\_c,m=c.unmount,s=c.\_\_;function d(n,t){c.\_\_h&&c.\_\_h(r,n,o||t),o=0;var u=r.\_\_H||(r.\_\_H={\_\_:[],\_\_h:[]});return n>=u.\_\_.length&&u.\_\_.push({}),u.\_\_[n]}function h(n){return o=1,p(D,n)}function p(n,u,i){var o=d(t++,2);if(o.t=n,!o.\_\_c&&(o.\_\_=[D(void 0,u),function(n){var t=o.\_\_N?o.\_\_N[0]:o.\_\_[0],r=o.t(t,n);t!==r&&(o.\_\_N=[r,o.\_\_[1]],o.\_\_c.setState({}));}],o.\_\_c=r,!r.u)){var f=function(n,t,r){if(!o.\_\_c.\_\_H)return !0;var u=o.\_\_c.\_\_H.\_\_.filter(function(n){return !!n.\_\_c});if(u.every(function(n){return !n.\_\_N}))return !c||c.call(this,n,t,r);var i=o.\_\_c.props!==n;return u.forEach(function(n){if(n.\_\_N){var t=n.\_\_[0];n.\_\_=n.\_\_N,n.\_\_N=void 0,t!==n.\_\_[0]&&(i=!0);}}),c&&c.call(this,n,t,r)||i};r.u=!0;var c=r.shouldComponentUpdate,e=r.componentWillUpdate;r.componentWillUpdate=function(n,t,r){if(this.\_\_e){var u=c;c=void 0,f(n,t,r),c=u;}e&&e.call(this,n,t,r);},r.shouldComponentUpdate=f;}return o.\_\_N||o.\_\_}function y(n,u){var i=d(t++,3);!c.\_\_s&&C(i.\_\_H,u)&&(i.\_\_=n,i.i=u,r.\_\_H.\_\_h.push(i));}function \_(n,u){var i=d(t++,4);!c.\_\_s&&C(i.\_\_H,u)&&(i.\_\_=n,i.i=u,r.\_\_h.push(i));}function A(n){return o=5,T(function(){return {current:n}},[])}function T(n,r){var u=d(t++,7);return C(u.\_\_H,r)&&(u.\_\_=n(),u.\_\_H=r,u.\_\_h=n),u.\_\_}function q(n,t){return o=8,T(function(){return n},t)}function x(n){var u=r.context[n.\_\_c],i=d(t++,9);return i.c=n,u?(null==i.\_\_&&(i.\_\_=!0,u.sub(r)),u.props.value):n.\_\_}function j(){for(var n;n=f.shift();)if(n.\_\_P&&n.\_\_H)try{n.\_\_H.\_\_h.forEach(z),n.\_\_H.\_\_h.forEach(B),n.\_\_H.\_\_h=[];}catch(t){n.\_\_H.\_\_h=[],c.\_\_e(t,n.\_\_v);}}c.\_\_b=function(n){r=null,e&&e(n);},c.\_\_=function(n,t){n&&t.\_\_k&&t.\_\_k.\_\_m&&(n.\_\_m=t.\_\_k.\_\_m),s&&s(n,t);},c.\_\_r=function(n){a&&a(n),t=0;var i=(r=n.\_\_c).\_\_H;i&&(u===r?(i.\_\_h=[],r.\_\_h=[],i.\_\_.forEach(function(n){n.\_\_N&&(n.\_\_=n.\_\_N),n.i=n.\_\_N=void 0;})):(i.\_\_h.forEach(z),i.\_\_h.forEach(B),i.\_\_h=[],t=0)),u=r;},c.diffed=function(n){v&&v(n);var t=n.\_\_c;t&&t.\_\_H&&(t.\_\_H.\_\_h.length&&(1!==f.push(t)&&i===c.requestAnimationFrame||((i=c.requestAnimationFrame)||w)(j)),t.\_\_H.\_\_.forEach(function(n){n.i&&(n.\_\_H=n.i),n.i=void 0;})),u=r=null;},c.\_\_c=function(n,t){t.some(function(n){try{n.\_\_h.forEach(z),n.\_\_h=n.\_\_h.filter(function(n){return !n.\_\_||B(n)});}catch(r){t.some(function(n){n.\_\_h&&(n.\_\_h=[]);}),t=[],c.\_\_e(r,n.\_\_v);}}),l&&l(n,t);},c.unmount=function(n){m&&m(n);var t,r=n.\_\_c;r&&r.\_\_H&&(r.\_\_H.\_\_.forEach(function(n){try{z(n);}catch(n){t=n;}}),r.\_\_H=void 0,t&&c.\_\_e(t,r.\_\_v));};var k="function"==typeof requestAnimationFrame;function w(n){var t,r=function(){clearTimeout(u),k&&cancelAnimationFrame(t),setTimeout(n);},u=setTimeout(r,100);k&&(t=requestAnimationFrame(r));}function z(n){var t=r,u=n.\_\_c;"function"==typeof u&&(n.\_\_c=void 0,u()),r=t;}function B(n){var t=r;n.\_\_c=n.\_\_(),r=t;}function C(n,t){return !n||n.length!==t.length||t.some(function(t,r){return t!==n[r]})}function D(n,t){return "function"==typeof t?t(n):t}

const PLACEHOLDER = "\*/\*\*/file.js";

const SideBar = ({ availableSizeProperties, sizeProperty, setSizeProperty, onExcludeChange, onIncludeChange, }) => {

const [includeValue, setIncludeValue] = h("");

const [excludeValue, setExcludeValue] = h("");

const handleSizePropertyChange = (sizeProp) => () => {

if (sizeProp !== sizeProperty) {

setSizeProperty(sizeProp);

}

};

const handleIncludeChange = (event) => {

const value = event.currentTarget.value;

setIncludeValue(value);

onIncludeChange(value);

};

const handleExcludeChange = (event) => {

const value = event.currentTarget.value;

setExcludeValue(value);

onExcludeChange(value);

};

return (u$1("aside", { className: "sidebar", children: [u$1("div", { className: "size-selectors", children: availableSizeProperties.length > 1 &&

availableSizeProperties.map((sizeProp) => {

const id = `selector-${sizeProp}`;

return (u$1("div", { className: "size-selector", children: [u$1("input", { type: "radio", id: id, checked: sizeProp === sizeProperty, onChange: handleSizePropertyChange(sizeProp) }), u$1("label", { htmlFor: id, children: LABELS[sizeProp] })] }, sizeProp));

}) }), u$1("div", { className: "module-filters", children: [u$1("div", { className: "module-filter", children: [u$1("label", { htmlFor: "module-filter-exclude", children: "Exclude" }), u$1("input", { type: "text", id: "module-filter-exclude", value: excludeValue, onInput: handleExcludeChange, placeholder: PLACEHOLDER })] }), u$1("div", { className: "module-filter", children: [u$1("label", { htmlFor: "module-filter-include", children: "Include" }), u$1("input", { type: "text", id: "module-filter-include", value: includeValue, onInput: handleIncludeChange, placeholder: PLACEHOLDER })] })] })] }));

};

function getDefaultExportFromCjs (x) {

return x && x.\_\_esModule && Object.prototype.hasOwnProperty.call(x, 'default') ? x['default'] : x;

}

var utils = {};

var constants$1;

var hasRequiredConstants;

function requireConstants () {

if (hasRequiredConstants) return constants$1;

hasRequiredConstants = 1;

const WIN\_SLASH = '\\\\/';

const WIN\_NO\_SLASH = `[^${WIN\_SLASH}]`;

/\*\*

\* Posix glob regex

\*/

const DOT\_LITERAL = '\\.';

const PLUS\_LITERAL = '\\+';

const QMARK\_LITERAL = '\\?';

const SLASH\_LITERAL = '\\/';

const ONE\_CHAR = '(?=.)';

const QMARK = '[^/]';

const END\_ANCHOR = `(?:${SLASH\_LITERAL}|$)`;

const START\_ANCHOR = `(?:^|${SLASH\_LITERAL})`;

const DOTS\_SLASH = `${DOT\_LITERAL}{1,2}${END\_ANCHOR}`;

const NO\_DOT = `(?!${DOT\_LITERAL})`;

const NO\_DOTS = `(?!${START\_ANCHOR}${DOTS\_SLASH})`;

const NO\_DOT\_SLASH = `(?!${DOT\_LITERAL}{0,1}${END\_ANCHOR})`;

const NO\_DOTS\_SLASH = `(?!${DOTS\_SLASH})`;

const QMARK\_NO\_DOT = `[^.${SLASH\_LITERAL}]`;

const STAR = `${QMARK}\*?`;

const SEP = '/';

const POSIX\_CHARS = {

DOT\_LITERAL,

PLUS\_LITERAL,

QMARK\_LITERAL,

SLASH\_LITERAL,

ONE\_CHAR,

QMARK,

END\_ANCHOR,

DOTS\_SLASH,

NO\_DOT,

NO\_DOTS,

NO\_DOT\_SLASH,

NO\_DOTS\_SLASH,

QMARK\_NO\_DOT,

STAR,

START\_ANCHOR,

SEP

};

/\*\*

\* Windows glob regex

\*/

const WINDOWS\_CHARS = {

...POSIX\_CHARS,

SLASH\_LITERAL: `[${WIN\_SLASH}]`,

QMARK: WIN\_NO\_SLASH,

STAR: `${WIN\_NO\_SLASH}\*?`,

DOTS\_SLASH: `${DOT\_LITERAL}{1,2}(?:[${WIN\_SLASH}]|$)`,

NO\_DOT: `(?!${DOT\_LITERAL})`,

NO\_DOTS: `(?!(?:^|[${WIN\_SLASH}])${DOT\_LITERAL}{1,2}(?:[${WIN\_SLASH}]|$))`,

NO\_DOT\_SLASH: `(?!${DOT\_LITERAL}{0,1}(?:[${WIN\_SLASH}]|$))`,

NO\_DOTS\_SLASH: `(?!${DOT\_LITERAL}{1,2}(?:[${WIN\_SLASH}]|$))`,

QMARK\_NO\_DOT: `[^.${WIN\_SLASH}]`,

START\_ANCHOR: `(?:^|[${WIN\_SLASH}])`,

END\_ANCHOR: `(?:[${WIN\_SLASH}]|$)`,

SEP: '\\'

};

/\*\*

\* POSIX Bracket Regex

\*/

const POSIX\_REGEX\_SOURCE = {

alnum: 'a-zA-Z0-9',

alpha: 'a-zA-Z',

ascii: '\\x00-\\x7F',

blank: ' \\t',

cntrl: '\\x00-\\x1F\\x7F',

digit: '0-9',

graph: '\\x21-\\x7E',

lower: 'a-z',

print: '\\x20-\\x7E ',

punct: '\\-!"#$%&\'()\\\*+,./:;<=>?@[\\]^\_`{|}~',

space: ' \\t\\r\\n\\v\\f',

upper: 'A-Z',

word: 'A-Za-z0-9\_',

xdigit: 'A-Fa-f0-9'

};

constants$1 = {

MAX\_LENGTH: 1024 \* 64,

POSIX\_REGEX\_SOURCE,

REGEX\_BACKSLASH: /\\(?![\*+?^${}(|)[\]])/g,

REGEX\_NON\_SPECIAL\_CHARS: /^[^@![\].,$\*+?^{}()|\\/]+/,

REGEX\_SPECIAL\_CHARS: /[-\*+?.^${}(|)[\]]/,

REGEX\_SPECIAL\_CHARS\_BACKREF: /(\\?)((\W)(\3\*))/g,

REGEX\_SPECIAL\_CHARS\_GLOBAL: /([-\*+?.^${}(|)[\]])/g,

REGEX\_REMOVE\_BACKSLASH: /(?:\[.\*?[^\\]\]|\\(?=.))/g,

REPLACEMENTS: {

'\*\*\*': '\*',

'\*\*/\*\*': '\*\*',

'\*\*/\*\*/\*\*': '\*\*'

},

CHAR\_0: 48, /\* 0 \*/

CHAR\_9: 57, /\* 9 \*/

CHAR\_UPPERCASE\_A: 65, /\* A \*/

CHAR\_LOWERCASE\_A: 97, /\* a \*/

CHAR\_UPPERCASE\_Z: 90, /\* Z \*/

CHAR\_LOWERCASE\_Z: 122, /\* z \*/

CHAR\_LEFT\_PARENTHESES: 40, /\* ( \*/

CHAR\_RIGHT\_PARENTHESES: 41, /\* ) \*/

CHAR\_ASTERISK: 42, /\* \* \*/

CHAR\_AMPERSAND: 38, /\* & \*/

CHAR\_AT: 64, /\* @ \*/

CHAR\_BACKWARD\_SLASH: 92, /\* \ \*/

CHAR\_CARRIAGE\_RETURN: 13, /\* \r \*/

CHAR\_CIRCUMFLEX\_ACCENT: 94, /\* ^ \*/

CHAR\_COLON: 58, /\* : \*/

CHAR\_COMMA: 44, /\* , \*/

CHAR\_DOT: 46, /\* . \*/

CHAR\_DOUBLE\_QUOTE: 34, /\* " \*/

CHAR\_EQUAL: 61, /\* = \*/

CHAR\_EXCLAMATION\_MARK: 33, /\* ! \*/

CHAR\_FORM\_FEED: 12, /\* \f \*/

CHAR\_FORWARD\_SLASH: 47, /\* / \*/

CHAR\_GRAVE\_ACCENT: 96, /\* ` \*/

CHAR\_HASH: 35, /\* # \*/

CHAR\_HYPHEN\_MINUS: 45, /\* - \*/

CHAR\_LEFT\_ANGLE\_BRACKET: 60, /\* < \*/

CHAR\_LEFT\_CURLY\_BRACE: 123, /\* { \*/

CHAR\_LEFT\_SQUARE\_BRACKET: 91, /\* [ \*/

CHAR\_LINE\_FEED: 10, /\* \n \*/

CHAR\_NO\_BREAK\_SPACE: 160, /\* \u00A0 \*/

CHAR\_PERCENT: 37, /\* % \*/

CHAR\_PLUS: 43, /\* + \*/

CHAR\_QUESTION\_MARK: 63, /\* ? \*/

CHAR\_RIGHT\_ANGLE\_BRACKET: 62, /\* > \*/

CHAR\_RIGHT\_CURLY\_BRACE: 125, /\* } \*/

CHAR\_RIGHT\_SQUARE\_BRACKET: 93, /\* ] \*/

CHAR\_SEMICOLON: 59, /\* ; \*/

CHAR\_SINGLE\_QUOTE: 39, /\* ' \*/

CHAR\_SPACE: 32, /\* \*/

CHAR\_TAB: 9, /\* \t \*/

CHAR\_UNDERSCORE: 95, /\* \_ \*/

CHAR\_VERTICAL\_LINE: 124, /\* | \*/

CHAR\_ZERO\_WIDTH\_NOBREAK\_SPACE: 65279, /\* \uFEFF \*/

/\*\*

\* Create EXTGLOB\_CHARS

\*/

extglobChars(chars) {

return {

'!': { type: 'negate', open: '(?:(?!(?:', close: `))${chars.STAR})` },

'?': { type: 'qmark', open: '(?:', close: ')?' },

'+': { type: 'plus', open: '(?:', close: ')+' },

'\*': { type: 'star', open: '(?:', close: ')\*' },

'@': { type: 'at', open: '(?:', close: ')' }

};

},

/\*\*

\* Create GLOB\_CHARS

\*/

globChars(win32) {

return win32 === true ? WINDOWS\_CHARS : POSIX\_CHARS;

}

};

return constants$1;

}

/\*global navigator\*/

var hasRequiredUtils;

function requireUtils () {

if (hasRequiredUtils) return utils;

hasRequiredUtils = 1;

(function (exports) {

const {

REGEX\_BACKSLASH,

REGEX\_REMOVE\_BACKSLASH,

REGEX\_SPECIAL\_CHARS,

REGEX\_SPECIAL\_CHARS\_GLOBAL

} = /\*@\_\_PURE\_\_\*/ requireConstants();

exports.isObject = val => val !== null && typeof val === 'object' && !Array.isArray(val);

exports.hasRegexChars = str => REGEX\_SPECIAL\_CHARS.test(str);

exports.isRegexChar = str => str.length === 1 && exports.hasRegexChars(str);

exports.escapeRegex = str => str.replace(REGEX\_SPECIAL\_CHARS\_GLOBAL, '\\$1');

exports.toPosixSlashes = str => str.replace(REGEX\_BACKSLASH, '/');

exports.isWindows = () => {

if (typeof navigator !== 'undefined' && navigator.platform) {

const platform = navigator.platform.toLowerCase();

return platform === 'win32' || platform === 'windows';

}

if (typeof process !== 'undefined' && process.platform) {

return process.platform === 'win32';

}

return false;

};

exports.removeBackslashes = str => {

return str.replace(REGEX\_REMOVE\_BACKSLASH, match => {

return match === '\\' ? '' : match;

});

};

exports.escapeLast = (input, char, lastIdx) => {

const idx = input.lastIndexOf(char, lastIdx);

if (idx === -1) return input;

if (input[idx - 1] === '\\') return exports.escapeLast(input, char, idx - 1);

return `${input.slice(0, idx)}\\${input.slice(idx)}`;

};

exports.removePrefix = (input, state = {}) => {

let output = input;

if (output.startsWith('./')) {

output = output.slice(2);

state.prefix = './';

}

return output;

};

exports.wrapOutput = (input, state = {}, options = {}) => {

const prepend = options.contains ? '' : '^';

const append = options.contains ? '' : '$';

let output = `${prepend}(?:${input})${append}`;

if (state.negated === true) {

output = `(?:^(?!${output}).\*$)`;

}

return output;

};

exports.basename = (path, { windows } = {}) => {

const segs = path.split(windows ? /[\\/]/ : '/');

const last = segs[segs.length - 1];

if (last === '') {

return segs[segs.length - 2];

}

return last;

};

} (utils));

return utils;

}

var scan\_1;

var hasRequiredScan;

function requireScan () {

if (hasRequiredScan) return scan\_1;

hasRequiredScan = 1;

const utils = /\*@\_\_PURE\_\_\*/ requireUtils();

const {

CHAR\_ASTERISK, /\* \* \*/

CHAR\_AT, /\* @ \*/

CHAR\_BACKWARD\_SLASH, /\* \ \*/

CHAR\_COMMA, /\* , \*/

CHAR\_DOT, /\* . \*/

CHAR\_EXCLAMATION\_MARK, /\* ! \*/

CHAR\_FORWARD\_SLASH, /\* / \*/

CHAR\_LEFT\_CURLY\_BRACE, /\* { \*/

CHAR\_LEFT\_PARENTHESES, /\* ( \*/

CHAR\_LEFT\_SQUARE\_BRACKET, /\* [ \*/

CHAR\_PLUS, /\* + \*/

CHAR\_QUESTION\_MARK, /\* ? \*/

CHAR\_RIGHT\_CURLY\_BRACE, /\* } \*/

CHAR\_RIGHT\_PARENTHESES, /\* ) \*/

CHAR\_RIGHT\_SQUARE\_BRACKET /\* ] \*/

} = /\*@\_\_PURE\_\_\*/ requireConstants();

const isPathSeparator = code => {

return code === CHAR\_FORWARD\_SLASH || code === CHAR\_BACKWARD\_SLASH;

};

const depth = token => {

if (token.isPrefix !== true) {

token.depth = token.isGlobstar ? Infinity : 1;

}

};

/\*\*

\* Quickly scans a glob pattern and returns an object with a handful of

\* useful properties, like `isGlob`, `path` (the leading non-glob, if it exists),

\* `glob` (the actual pattern), `negated` (true if the path starts with `!` but not

\* with `!(`) and `negatedExtglob` (true if the path starts with `!(`).

\*

\* ```js

\* const pm = require('picomatch');

\* console.log(pm.scan('foo/bar/\*.js'));

\* { isGlob: true, input: 'foo/bar/\*.js', base: 'foo/bar', glob: '\*.js' }

\* ```

\* @param {String} `str`

\* @param {Object} `options`

\* @return {Object} Returns an object with tokens and regex source string.

\* @api public

\*/

const scan = (input, options) => {

const opts = options || {};

const length = input.length - 1;

const scanToEnd = opts.parts === true || opts.scanToEnd === true;

const slashes = [];

const tokens = [];

const parts = [];

let str = input;

let index = -1;

let start = 0;

let lastIndex = 0;

let isBrace = false;

let isBracket = false;

let isGlob = false;

let isExtglob = false;

let isGlobstar = false;

let braceEscaped = false;

let backslashes = false;

let negated = false;

let negatedExtglob = false;

let finished = false;

let braces = 0;

let prev;

let code;

let token = { value: '', depth: 0, isGlob: false };

const eos = () => index >= length;

const peek = () => str.charCodeAt(index + 1);

const advance = () => {

prev = code;

return str.charCodeAt(++index);

};

while (index < length) {

code = advance();

let next;

if (code === CHAR\_BACKWARD\_SLASH) {

backslashes = token.backslashes = true;

code = advance();

if (code === CHAR\_LEFT\_CURLY\_BRACE) {

braceEscaped = true;

}

continue;

}

if (braceEscaped === true || code === CHAR\_LEFT\_CURLY\_BRACE) {

braces++;

while (eos() !== true && (code = advance())) {

if (code === CHAR\_BACKWARD\_SLASH) {

backslashes = token.backslashes = true;

advance();

continue;

}

if (code === CHAR\_LEFT\_CURLY\_BRACE) {

braces++;

continue;

}

if (braceEscaped !== true && code === CHAR\_DOT && (code = advance()) === CHAR\_DOT) {

isBrace = token.isBrace = true;

isGlob = token.isGlob = true;

finished = true;

if (scanToEnd === true) {

continue;

}

break;

}

if (braceEscaped !== true && code === CHAR\_COMMA) {

isBrace = token.isBrace = true;

isGlob = token.isGlob = true;

finished = true;

if (scanToEnd === true) {

continue;

}

break;

}

if (code === CHAR\_RIGHT\_CURLY\_BRACE) {

braces--;

if (braces === 0) {

braceEscaped = false;

isBrace = token.isBrace = true;

finished = true;

break;

}

}

}

if (scanToEnd === true) {

continue;

}

break;

}

if (code === CHAR\_FORWARD\_SLASH) {

slashes.push(index);

tokens.push(token);

token = { value: '', depth: 0, isGlob: false };

if (finished === true) continue;

if (prev === CHAR\_DOT && index === (start + 1)) {

start += 2;

continue;

}

lastIndex = index + 1;

continue;

}

if (opts.noext !== true) {

const isExtglobChar = code === CHAR\_PLUS

|| code === CHAR\_AT

|| code === CHAR\_ASTERISK

|| code === CHAR\_QUESTION\_MARK

|| code === CHAR\_EXCLAMATION\_MARK;

if (isExtglobChar === true && peek() === CHAR\_LEFT\_PARENTHESES) {

isGlob = token.isGlob = true;

isExtglob = token.isExtglob = true;

finished = true;

if (code === CHAR\_EXCLAMATION\_MARK && index === start) {

negatedExtglob = true;

}

if (scanToEnd === true) {

while (eos() !== true && (code = advance())) {

if (code === CHAR\_BACKWARD\_SLASH) {

backslashes = token.backslashes = true;

code = advance();

continue;

}

if (code === CHAR\_RIGHT\_PARENTHESES) {

isGlob = token.isGlob = true;

finished = true;

break;

}

}

continue;

}

break;

}

}

if (code === CHAR\_ASTERISK) {

if (prev === CHAR\_ASTERISK) isGlobstar = token.isGlobstar = true;

isGlob = token.isGlob = true;

finished = true;

if (scanToEnd === true) {

continue;

}

break;

}

if (code === CHAR\_QUESTION\_MARK) {

isGlob = token.isGlob = true;

finished = true;

if (scanToEnd === true) {

continue;

}

break;

}

if (code === CHAR\_LEFT\_SQUARE\_BRACKET) {

while (eos() !== true && (next = advance())) {

if (next === CHAR\_BACKWARD\_SLASH) {

backslashes = token.backslashes = true;

advance();

continue;

}

if (next === CHAR\_RIGHT\_SQUARE\_BRACKET) {

isBracket = token.isBracket = true;

isGlob = token.isGlob = true;

finished = true;

break;

}

}

if (scanToEnd === true) {

continue;

}

break;

}

if (opts.nonegate !== true && code === CHAR\_EXCLAMATION\_MARK && index === start) {

negated = token.negated = true;

start++;

continue;

}

if (opts.noparen !== true && code === CHAR\_LEFT\_PARENTHESES) {

isGlob = token.isGlob = true;

if (scanToEnd === true) {

while (eos() !== true && (code = advance())) {

if (code === CHAR\_LEFT\_PARENTHESES) {

backslashes = token.backslashes = true;

code = advance();

continue;

}

if (code === CHAR\_RIGHT\_PARENTHESES) {

finished = true;

break;

}

}

continue;

}

break;

}

if (isGlob === true) {

finished = true;

if (scanToEnd === true) {

continue;

}

break;

}

}

if (opts.noext === true) {

isExtglob = false;

isGlob = false;

}

let base = str;

let prefix = '';

let glob = '';

if (start > 0) {

prefix = str.slice(0, start);

str = str.slice(start);

lastIndex -= start;

}

if (base && isGlob === true && lastIndex > 0) {

base = str.slice(0, lastIndex);

glob = str.slice(lastIndex);

} else if (isGlob === true) {

base = '';

glob = str;

} else {

base = str;

}

if (base && base !== '' && base !== '/' && base !== str) {

if (isPathSeparator(base.charCodeAt(base.length - 1))) {

base = base.slice(0, -1);

}

}

if (opts.unescape === true) {

if (glob) glob = utils.removeBackslashes(glob);

if (base && backslashes === true) {

base = utils.removeBackslashes(base);

}

}

const state = {

prefix,

input,

start,

base,

glob,

isBrace,

isBracket,

isGlob,

isExtglob,

isGlobstar,

negated,

negatedExtglob

};

if (opts.tokens === true) {

state.maxDepth = 0;

if (!isPathSeparator(code)) {

tokens.push(token);

}

state.tokens = tokens;

}

if (opts.parts === true || opts.tokens === true) {

let prevIndex;

for (let idx = 0; idx < slashes.length; idx++) {

const n = prevIndex ? prevIndex + 1 : start;

const i = slashes[idx];

const value = input.slice(n, i);

if (opts.tokens) {

if (idx === 0 && start !== 0) {

tokens[idx].isPrefix = true;

tokens[idx].value = prefix;

} else {

tokens[idx].value = value;

}

depth(tokens[idx]);

state.maxDepth += tokens[idx].depth;

}

if (idx !== 0 || value !== '') {

parts.push(value);

}

prevIndex = i;

}

if (prevIndex && prevIndex + 1 < input.length) {

const value = input.slice(prevIndex + 1);

parts.push(value);

if (opts.tokens) {

tokens[tokens.length - 1].value = value;

depth(tokens[tokens.length - 1]);

state.maxDepth += tokens[tokens.length - 1].depth;

}

}

state.slashes = slashes;

state.parts = parts;

}

return state;

};

scan\_1 = scan;

return scan\_1;

}

var parse\_1;

var hasRequiredParse;

function requireParse () {

if (hasRequiredParse) return parse\_1;

hasRequiredParse = 1;

const constants = /\*@\_\_PURE\_\_\*/ requireConstants();

const utils = /\*@\_\_PURE\_\_\*/ requireUtils();

/\*\*

\* Constants

\*/

const {

MAX\_LENGTH,

POSIX\_REGEX\_SOURCE,

REGEX\_NON\_SPECIAL\_CHARS,

REGEX\_SPECIAL\_CHARS\_BACKREF,

REPLACEMENTS

} = constants;

/\*\*

\* Helpers

\*/

const expandRange = (args, options) => {

if (typeof options.expandRange === 'function') {

return options.expandRange(...args, options);

}

args.sort();

const value = `[${args.join('-')}]`;

try {

/\* eslint-disable-next-line no-new \*/

new RegExp(value);

} catch (ex) {

return args.map(v => utils.escapeRegex(v)).join('..');

}

return value;

};

/\*\*

\* Create the message for a syntax error

\*/

const syntaxError = (type, char) => {

return `Missing ${type}: "${char}" - use "\\\\${char}" to match literal characters`;

};

/\*\*

\* Parse the given input string.

\* @param {String} input

\* @param {Object} options

\* @return {Object}

\*/

const parse = (input, options) => {

if (typeof input !== 'string') {

throw new TypeError('Expected a string');

}

input = REPLACEMENTS[input] || input;

const opts = { ...options };

const max = typeof opts.maxLength === 'number' ? Math.min(MAX\_LENGTH, opts.maxLength) : MAX\_LENGTH;

let len = input.length;

if (len > max) {

throw new SyntaxError(`Input length: ${len}, exceeds maximum allowed length: ${max}`);

}

const bos = { type: 'bos', value: '', output: opts.prepend || '' };

const tokens = [bos];

const capture = opts.capture ? '' : '?:';

const PLATFORM\_CHARS = constants.globChars(opts.windows);

const EXTGLOB\_CHARS = constants.extglobChars(PLATFORM\_CHARS);

const {

DOT\_LITERAL,

PLUS\_LITERAL,

SLASH\_LITERAL,

ONE\_CHAR,

DOTS\_SLASH,

NO\_DOT,

NO\_DOT\_SLASH,

NO\_DOTS\_SLASH,

QMARK,

QMARK\_NO\_DOT,

STAR,

START\_ANCHOR

} = PLATFORM\_CHARS;

const globstar = opts => {

return `(${capture}(?:(?!${START\_ANCHOR}${opts.dot ? DOTS\_SLASH : DOT\_LITERAL}).)\*?)`;

};

const nodot = opts.dot ? '' : NO\_DOT;

const qmarkNoDot = opts.dot ? QMARK : QMARK\_NO\_DOT;

let star = opts.bash === true ? globstar(opts) : STAR;

if (opts.capture) {

star = `(${star})`;

}

if (typeof opts.noext === 'boolean') {

opts.noextglob = opts.noext;

}

const state = {

input,

index: -1,

start: 0,

dot: opts.dot === true,

consumed: '',

output: '',

prefix: '',

backtrack: false,

negated: false,

brackets: 0,

braces: 0,

parens: 0,

quotes: 0,

globstar: false,

tokens

};

input = utils.removePrefix(input, state);

len = input.length;

const extglobs = [];

const braces = [];

const stack = [];

let prev = bos;

let value;

/\*\*

\* Tokenizing helpers

\*/

const eos = () => state.index === len - 1;

const peek = state.peek = (n = 1) => input[state.index + n];

const advance = state.advance = () => input[++state.index] || '';

const remaining = () => input.slice(state.index + 1);

const consume = (value = '', num = 0) => {

state.consumed += value;

state.index += num;

};

const append = token => {

state.output += token.output != null ? token.output : token.value;

consume(token.value);

};

const negate = () => {

let count = 1;

while (peek() === '!' && (peek(2) !== '(' || peek(3) === '?')) {

advance();

state.start++;

count++;

}

if (count % 2 === 0) {

return false;

}

state.negated = true;

state.start++;

return true;

};

const increment = type => {

state[type]++;

stack.push(type);

};

const decrement = type => {

state[type]--;

stack.pop();

};

/\*\*

\* Push tokens onto the tokens array. This helper speeds up

\* tokenizing by 1) helping us avoid backtracking as much as possible,

\* and 2) helping us avoid creating extra tokens when consecutive

\* characters are plain text. This improves performance and simplifies

\* lookbehinds.

\*/

const push = tok => {

if (prev.type === 'globstar') {

const isBrace = state.braces > 0 && (tok.type === 'comma' || tok.type === 'brace');

const isExtglob = tok.extglob === true || (extglobs.length && (tok.type === 'pipe' || tok.type === 'paren'));

if (tok.type !== 'slash' && tok.type !== 'paren' && !isBrace && !isExtglob) {

state.output = state.output.slice(0, -prev.output.length);

prev.type = 'star';

prev.value = '\*';

prev.output = star;

state.output += prev.output;

}

}

if (extglobs.length && tok.type !== 'paren') {

extglobs[extglobs.length - 1].inner += tok.value;

}

if (tok.value || tok.output) append(tok);

if (prev && prev.type === 'text' && tok.type === 'text') {

prev.output = (prev.output || prev.value) + tok.value;

prev.value += tok.value;

return;

}

tok.prev = prev;

tokens.push(tok);

prev = tok;

};

const extglobOpen = (type, value) => {

const token = { ...EXTGLOB\_CHARS[value], conditions: 1, inner: '' };

token.prev = prev;

token.parens = state.parens;

token.output = state.output;

const output = (opts.capture ? '(' : '') + token.open;

increment('parens');

push({ type, value, output: state.output ? '' : ONE\_CHAR });

push({ type: 'paren', extglob: true, value: advance(), output });

extglobs.push(token);

};

const extglobClose = token => {

let output = token.close + (opts.capture ? ')' : '');

let rest;

if (token.type === 'negate') {

let extglobStar = star;

if (token.inner && token.inner.length > 1 && token.inner.includes('/')) {

extglobStar = globstar(opts);

}

if (extglobStar !== star || eos() || /^\)+$/.test(remaining())) {

output = token.close = `)$))${extglobStar}`;

}

if (token.inner.includes('\*') && (rest = remaining()) && /^\.[^\\/.]+$/.test(rest)) {

const expression = parse(rest, { ...options, fastpaths: false }).output;

output = token.close = `)${expression})${extglobStar})`;

}

if (token.prev.type === 'bos') {

state.negatedExtglob = true;

}

}

push({ type: 'paren', extglob: true, value, output });

decrement('parens');

};

/\*\*

\* Fast paths

\*/

if (opts.fastpaths !== false && !/(^[\*!]|[/()[\]{}"])/.test(input)) {

let backslashes = false;

let output = input.replace(REGEX\_SPECIAL\_CHARS\_BACKREF, (m, esc, chars, first, rest, index) => {

if (first === '\\') {

backslashes = true;

return m;

}

if (first === '?') {

if (esc) {

return esc + first + (rest ? QMARK.repeat(rest.length) : '');

}

if (index === 0) {

return qmarkNoDot + (rest ? QMARK.repeat(rest.length) : '');

}

return QMARK.repeat(chars.length);

}

if (first === '.') {

return DOT\_LITERAL.repeat(chars.length);

}

if (first === '\*') {

if (esc) {

return esc + first + (rest ? star : '');

}

return star;

}

return esc ? m : `\\${m}`;

});

if (backslashes === true) {

if (opts.unescape === true) {

output = output.replace(/\\/g, '');

} else {

output = output.replace(/\\+/g, m => {

return m.length % 2 === 0 ? '\\\\' : (m ? '\\' : '');

});

}

}

if (output === input && opts.contains === true) {

state.output = input;

return state;

}

state.output = utils.wrapOutput(output, state, options);

return state;

}

/\*\*

\* Tokenize input until we reach end-of-string

\*/

while (!eos()) {

value = advance();

if (value === '\u0000') {

continue;

}

/\*\*

\* Escaped characters

\*/

if (value === '\\') {

const next = peek();

if (next === '/' && opts.bash !== true) {

continue;

}

if (next === '.' || next === ';') {

continue;

}

if (!next) {

value += '\\';

push({ type: 'text', value });

continue;

}

const match = /^\\+/.exec(remaining());

let slashes = 0;

if (match && match[0].length > 2) {

slashes = match[0].length;

state.index += slashes;

if (slashes % 2 !== 0) {

value += '\\';

}

}

if (opts.unescape === true) {

value = advance();

} else {

value += advance();

}

if (state.brackets === 0) {

push({ type: 'text', value });

continue;

}

}

/\*\*

\* If we're inside a regex character class, continue

\* until we reach the closing bracket.

\*/

if (state.brackets > 0 && (value !== ']' || prev.value === '[' || prev.value === '[^')) {

if (opts.posix !== false && value === ':') {

const inner = prev.value.slice(1);

if (inner.includes('[')) {

prev.posix = true;

if (inner.includes(':')) {

const idx = prev.value.lastIndexOf('[');

const pre = prev.value.slice(0, idx);

const rest = prev.value.slice(idx + 2);

const posix = POSIX\_REGEX\_SOURCE[rest];

if (posix) {

prev.value = pre + posix;

state.backtrack = true;

advance();

if (!bos.output && tokens.indexOf(prev) === 1) {

bos.output = ONE\_CHAR;

}

continue;

}

}

}

}

if ((value === '[' && peek() !== ':') || (value === '-' && peek() === ']')) {

value = `\\${value}`;

}

if (value === ']' && (prev.value === '[' || prev.value === '[^')) {

value = `\\${value}`;

}

if (opts.posix === true && value === '!' && prev.value === '[') {

value = '^';

}

prev.value += value;

append({ value });

continue;

}

/\*\*

\* If we're inside a quoted string, continue

\* until we reach the closing double quote.

\*/

if (state.quotes === 1 && value !== '"') {

value = utils.escapeRegex(value);

prev.value += value;

append({ value });

continue;

}

/\*\*

\* Double quotes

\*/

if (value === '"') {

state.quotes = state.quotes === 1 ? 0 : 1;

if (opts.keepQuotes === true) {

push({ type: 'text', value });

}

continue;

}

/\*\*

\* Parentheses

\*/

if (value === '(') {

increment('parens');

push({ type: 'paren', value });

continue;

}

if (value === ')') {

if (state.parens === 0 && opts.strictBrackets === true) {

throw new SyntaxError(syntaxError('opening', '('));

}

const extglob = extglobs[extglobs.length - 1];

if (extglob && state.parens === extglob.parens + 1) {

extglobClose(extglobs.pop());

continue;

}

push({ type: 'paren', value, output: state.parens ? ')' : '\\)' });

decrement('parens');

continue;

}

/\*\*

\* Square brackets

\*/

if (value === '[') {

if (opts.nobracket === true || !remaining().includes(']')) {

if (opts.nobracket !== true && opts.strictBrackets === true) {

throw new SyntaxError(syntaxError('closing', ']'));

}

value = `\\${value}`;

} else {

increment('brackets');

}

push({ type: 'bracket', value });

continue;

}

if (value === ']') {

if (opts.nobracket === true || (prev && prev.type === 'bracket' && prev.value.length === 1)) {

push({ type: 'text', value, output: `\\${value}` });

continue;

}

if (state.brackets === 0) {

if (opts.strictBrackets === true) {

throw new SyntaxError(syntaxError('opening', '['));

}

push({ type: 'text', value, output: `\\${value}` });

continue;

}

decrement('brackets');

const prevValue = prev.value.slice(1);

if (prev.posix !== true && prevValue[0] === '^' && !prevValue.includes('/')) {

value = `/${value}`;

}

prev.value += value;

append({ value });

if (opts.literalBrackets === false || utils.hasRegexChars(prevValue)) {

continue;

}

const escaped = utils.escapeRegex(prev.value);

state.output = state.output.slice(0, -prev.value.length);

if (opts.literalBrackets === true) {

state.output += escaped;

prev.value = escaped;

continue;

}

prev.value = `(${capture}${escaped}|${prev.value})`;

state.output += prev.value;

continue;

}

/\*\*

\* Braces

\*/

if (value === '{' && opts.nobrace !== true) {

increment('braces');

const open = {

type: 'brace',

value,

output: '(',

outputIndex: state.output.length,

tokensIndex: state.tokens.length

};

braces.push(open);

push(open);

continue;

}

if (value === '}') {

const brace = braces[braces.length - 1];

if (opts.nobrace === true || !brace) {

push({ type: 'text', value, output: value });

continue;

}

let output = ')';

if (brace.dots === true) {

const arr = tokens.slice();

const range = [];

for (let i = arr.length - 1; i >= 0; i--) {

tokens.pop();

if (arr[i].type === 'brace') {

break;

}

if (arr[i].type !== 'dots') {

range.unshift(arr[i].value);

}

}

output = expandRange(range, opts);

state.backtrack = true;

}

if (brace.comma !== true && brace.dots !== true) {

const out = state.output.slice(0, brace.outputIndex);

const toks = state.tokens.slice(brace.tokensIndex);

brace.value = brace.output = '\\{';

value = output = '\\}';

state.output = out;

for (const t of toks) {

state.output += (t.output || t.value);

}

}

push({ type: 'brace', value, output });

decrement('braces');

braces.pop();

continue;

}

/\*\*

\* Pipes

\*/

if (value === '|') {

if (extglobs.length > 0) {

extglobs[extglobs.length - 1].conditions++;

}

push({ type: 'text', value });

continue;

}

/\*\*

\* Commas

\*/

if (value === ',') {

let output = value;

const brace = braces[braces.length - 1];

if (brace && stack[stack.length - 1] === 'braces') {

brace.comma = true;

output = '|';

}

push({ type: 'comma', value, output });

continue;

}

/\*\*

\* Slashes

\*/

if (value === '/') {

if (prev.type === 'dot' && state.index === state.start + 1) {

state.start = state.index + 1;

state.consumed = '';

state.output = '';

tokens.pop();

prev = bos; // reset "prev" to the first token

continue;

}

push({ type: 'slash', value, output: SLASH\_LITERAL });

continue;

}

/\*\*

\* Dots

\*/

if (value === '.') {

if (state.braces > 0 && prev.type === 'dot') {

if (prev.value === '.') prev.output = DOT\_LITERAL;

const brace = braces[braces.length - 1];

prev.type = 'dots';

prev.output += value;

prev.value += value;

brace.dots = true;

continue;

}

if ((state.braces + state.parens) === 0 && prev.type !== 'bos' && prev.type !== 'slash') {

push({ type: 'text', value, output: DOT\_LITERAL });

continue;

}

push({ type: 'dot', value, output: DOT\_LITERAL });

continue;

}

/\*\*

\* Question marks

\*/

if (value === '?') {

const isGroup = prev && prev.value === '(';

if (!isGroup && opts.noextglob !== true && peek() === '(' && peek(2) !== '?') {

extglobOpen('qmark', value);

continue;

}

if (prev && prev.type === 'paren') {

const next = peek();

let output = value;

if ((prev.value === '(' && !/[!=<:]/.test(next)) || (next === '<' && !/<([!=]|\w+>)/.test(remaining()))) {

output = `\\${value}`;

}

push({ type: 'text', value, output });

continue;

}

if (opts.dot !== true && (prev.type === 'slash' || prev.type === 'bos')) {

push({ type: 'qmark', value, output: QMARK\_NO\_DOT });

continue;

}

push({ type: 'qmark', value, output: QMARK });

continue;

}

/\*\*

\* Exclamation

\*/

if (value === '!') {

if (opts.noextglob !== true && peek() === '(') {

if (peek(2) !== '?' || !/[!=<:]/.test(peek(3))) {

extglobOpen('negate', value);

continue;

}

}

if (opts.nonegate !== true && state.index === 0) {

negate();

continue;

}

}

/\*\*

\* Plus

\*/

if (value === '+') {

if (opts.noextglob !== true && peek() === '(' && peek(2) !== '?') {

extglobOpen('plus', value);

continue;

}

if ((prev && prev.value === '(') || opts.regex === false) {

push({ type: 'plus', value, output: PLUS\_LITERAL });

continue;

}

if ((prev && (prev.type === 'bracket' || prev.type === 'paren' || prev.type === 'brace')) || state.parens > 0) {

push({ type: 'plus', value });

continue;

}

push({ type: 'plus', value: PLUS\_LITERAL });

continue;

}

/\*\*

\* Plain text

\*/

if (value === '@') {

if (opts.noextglob !== true && peek() === '(' && peek(2) !== '?') {

push({ type: 'at', extglob: true, value, output: '' });

continue;

}

push({ type: 'text', value });

continue;

}

/\*\*

\* Plain text

\*/

if (value !== '\*') {

if (value === '$' || value === '^') {

value = `\\${value}`;

}

const match = REGEX\_NON\_SPECIAL\_CHARS.exec(remaining());

if (match) {

value += match[0];

state.index += match[0].length;

}

push({ type: 'text', value });

continue;

}

/\*\*

\* Stars

\*/

if (prev && (prev.type === 'globstar' || prev.star === true)) {

prev.type = 'star';

prev.star = true;

prev.value += value;

prev.output = star;

state.backtrack = true;

state.globstar = true;

consume(value);

continue;

}

let rest = remaining();

if (opts.noextglob !== true && /^\([^?]/.test(rest)) {

extglobOpen('star', value);

continue;

}

if (prev.type === 'star') {

if (opts.noglobstar === true) {

consume(value);

continue;

}

const prior = prev.prev;

const before = prior.prev;

const isStart = prior.type === 'slash' || prior.type === 'bos';

const afterStar = before && (before.type === 'star' || before.type === 'globstar');

if (opts.bash === true && (!isStart || (rest[0] && rest[0] !== '/'))) {

push({ type: 'star', value, output: '' });

continue;

}

const isBrace = state.braces > 0 && (prior.type === 'comma' || prior.type === 'brace');

const isExtglob = extglobs.length && (prior.type === 'pipe' || prior.type === 'paren');

if (!isStart && prior.type !== 'paren' && !isBrace && !isExtglob) {

push({ type: 'star', value, output: '' });

continue;

}

while (rest.slice(0, 3) === '/\*\*') {

const after = input[state.index + 4];

if (after && after !== '/') {

break;

}

rest = rest.slice(3);

consume('/\*\*', 3);

}

if (prior.type === 'bos' && eos()) {

prev.type = 'globstar';

prev.value += value;

prev.output = globstar(opts);

state.output = prev.output;

state.globstar = true;

consume(value);

continue;

}

if (prior.type === 'slash' && prior.prev.type !== 'bos' && !afterStar && eos()) {

state.output = state.output.slice(0, -(prior.output + prev.output).length);

prior.output = `(?:${prior.output}`;

prev.type = 'globstar';

prev.output = globstar(opts) + (opts.strictSlashes ? ')' : '|$)');

prev.value += value;

state.globstar = true;

state.output += prior.output + prev.output;

consume(value);

continue;

}

if (prior.type === 'slash' && prior.prev.type !== 'bos' && rest[0] === '/') {

const end = rest[1] !== void 0 ? '|$' : '';

state.output = state.output.slice(0, -(prior.output + prev.output).length);

prior.output = `(?:${prior.output}`;

prev.type = 'globstar';

prev.output = `${globstar(opts)}${SLASH\_LITERAL}|${SLASH\_LITERAL}${end})`;

prev.value += value;

state.output += prior.output + prev.output;

state.globstar = true;

consume(value + advance());

push({ type: 'slash', value: '/', output: '' });

continue;

}

if (prior.type === 'bos' && rest[0] === '/') {

prev.type = 'globstar';

prev.value += value;

prev.output = `(?:^|${SLASH\_LITERAL}|${globstar(opts)}${SLASH\_LITERAL})`;

state.output = prev.output;

state.globstar = true;

consume(value + advance());

push({ type: 'slash', value: '/', output: '' });

continue;

}

state.output = state.output.slice(0, -prev.output.length);

prev.type = 'globstar';

prev.output = globstar(opts);

prev.value += value;

state.output += prev.output;

state.globstar = true;

consume(value);

continue;

}

const token = { type: 'star', value, output: star };

if (opts.bash === true) {

token.output = '.\*?';

if (prev.type === 'bos' || prev.type === 'slash') {

token.output = nodot + token.output;

}

push(token);

continue;

}

if (prev && (prev.type === 'bracket' || prev.type === 'paren') && opts.regex === true) {

token.output = value;

push(token);

continue;

}

if (state.index === state.start || prev.type === 'slash' || prev.type === 'dot') {

if (prev.type === 'dot') {

state.output += NO\_DOT\_SLASH;

prev.output += NO\_DOT\_SLASH;

} else if (opts.dot === true) {

state.output += NO\_DOTS\_SLASH;

prev.output += NO\_DOTS\_SLASH;

} else {

state.output += nodot;

prev.output += nodot;

}

if (peek() !== '\*') {

state.output += ONE\_CHAR;

prev.output += ONE\_CHAR;

}

}

push(token);

}

while (state.brackets > 0) {

if (opts.strictBrackets === true) throw new SyntaxError(syntaxError('closing', ']'));

state.output = utils.escapeLast(state.output, '[');

decrement('brackets');

}

while (state.parens > 0) {

if (opts.strictBrackets === true) throw new SyntaxError(syntaxError('closing', ')'));

state.output = utils.escapeLast(state.output, '(');

decrement('parens');

}

while (state.braces > 0) {

if (opts.strictBrackets === true) throw new SyntaxError(syntaxError('closing', '}'));

state.output = utils.escapeLast(state.output, '{');

decrement('braces');

}

if (opts.strictSlashes !== true && (prev.type === 'star' || prev.type === 'bracket')) {

push({ type: 'maybe\_slash', value: '', output: `${SLASH\_LITERAL}?` });

}

if (state.backtrack === true) {

state.output = '';

for (const token of state.tokens) {

state.output += token.output != null ? token.output : token.value;

if (token.suffix) {

state.output += token.suffix;

}

}

}

return state;

};

/\*\*

\* Fast paths for creating regular expressions for common glob patterns.

\* This can significantly speed up processing and has very little downside

\* impact when none of the fast paths match.

\*/

parse.fastpaths = (input, options) => {

const opts = { ...options };

const max = typeof opts.maxLength === 'number' ? Math.min(MAX\_LENGTH, opts.maxLength) : MAX\_LENGTH;

const len = input.length;

if (len > max) {

throw new SyntaxError(`Input length: ${len}, exceeds maximum allowed length: ${max}`);

}

input = REPLACEMENTS[input] || input;

const {

DOT\_LITERAL,

SLASH\_LITERAL,

ONE\_CHAR,

DOTS\_SLASH,

NO\_DOT,

NO\_DOTS,

NO\_DOTS\_SLASH,

STAR,

START\_ANCHOR

} = constants.globChars(opts.windows);

const nodot = opts.dot ? NO\_DOTS : NO\_DOT;

const slashDot = opts.dot ? NO\_DOTS\_SLASH : NO\_DOT;

const capture = opts.capture ? '' : '?:';

const state = { negated: false, prefix: '' };

let star = opts.bash === true ? '.\*?' : STAR;

if (opts.capture) {

star = `(${star})`;

}

const globstar = opts => {

if (opts.noglobstar === true) return star;

return `(${capture}(?:(?!${START\_ANCHOR}${opts.dot ? DOTS\_SLASH : DOT\_LITERAL}).)\*?)`;

};

const create = str => {

switch (str) {

case '\*':

return `${nodot}${ONE\_CHAR}${star}`;

case '.\*':

return `${DOT\_LITERAL}${ONE\_CHAR}${star}`;

case '\*.\*':

return `${nodot}${star}${DOT\_LITERAL}${ONE\_CHAR}${star}`;

case '\*/\*':

return `${nodot}${star}${SLASH\_LITERAL}${ONE\_CHAR}${slashDot}${star}`;

case '\*\*':

return nodot + globstar(opts);

case '\*\*/\*':

return `(?:${nodot}${globstar(opts)}${SLASH\_LITERAL})?${slashDot}${ONE\_CHAR}${star}`;

case '\*\*/\*.\*':

return `(?:${nodot}${globstar(opts)}${SLASH\_LITERAL})?${slashDot}${star}${DOT\_LITERAL}${ONE\_CHAR}${star}`;

case '\*\*/.\*':

return `(?:${nodot}${globstar(opts)}${SLASH\_LITERAL})?${DOT\_LITERAL}${ONE\_CHAR}${star}`;

default: {

const match = /^(.\*?)\.(\w+)$/.exec(str);

if (!match) return;

const source = create(match[1]);

if (!source) return;

return source + DOT\_LITERAL + match[2];

}

}

};

const output = utils.removePrefix(input, state);

let source = create(output);

if (source && opts.strictSlashes !== true) {

source += `${SLASH\_LITERAL}?`;

}

return source;

};

parse\_1 = parse;

return parse\_1;

}

var picomatch\_1$1;

var hasRequiredPicomatch$1;

function requirePicomatch$1 () {

if (hasRequiredPicomatch$1) return picomatch\_1$1;

hasRequiredPicomatch$1 = 1;

const scan = /\*@\_\_PURE\_\_\*/ requireScan();

const parse = /\*@\_\_PURE\_\_\*/ requireParse();

const utils = /\*@\_\_PURE\_\_\*/ requireUtils();

const constants = /\*@\_\_PURE\_\_\*/ requireConstants();

const isObject = val => val && typeof val === 'object' && !Array.isArray(val);

/\*\*

\* Creates a matcher function from one or more glob patterns. The

\* returned function takes a string to match as its first argument,

\* and returns true if the string is a match. The returned matcher

\* function also takes a boolean as the second argument that, when true,

\* returns an object with additional information.

\*

\* ```js

\* const picomatch = require('picomatch');

\* // picomatch(glob[, options]);

\*

\* const isMatch = picomatch('\*.!(\*a)');

\* console.log(isMatch('a.a')); //=> false

\* console.log(isMatch('a.b')); //=> true

\* ```

\* @name picomatch

\* @param {String|Array} `globs` One or more glob patterns.

\* @param {Object=} `options`

\* @return {Function=} Returns a matcher function.

\* @api public

\*/

const picomatch = (glob, options, returnState = false) => {

if (Array.isArray(glob)) {

const fns = glob.map(input => picomatch(input, options, returnState));

const arrayMatcher = str => {

for (const isMatch of fns) {

const state = isMatch(str);

if (state) return state;

}

return false;

};

return arrayMatcher;

}

const isState = isObject(glob) && glob.tokens && glob.input;

if (glob === '' || (typeof glob !== 'string' && !isState)) {

throw new TypeError('Expected pattern to be a non-empty string');

}

const opts = options || {};

const posix = opts.windows;

const regex = isState

? picomatch.compileRe(glob, options)

: picomatch.makeRe(glob, options, false, true);

const state = regex.state;

delete regex.state;

let isIgnored = () => false;

if (opts.ignore) {

const ignoreOpts = { ...options, ignore: null, onMatch: null, onResult: null };

isIgnored = picomatch(opts.ignore, ignoreOpts, returnState);

}

const matcher = (input, returnObject = false) => {

const { isMatch, match, output } = picomatch.test(input, regex, options, { glob, posix });

const result = { glob, state, regex, posix, input, output, match, isMatch };

if (typeof opts.onResult === 'function') {

opts.onResult(result);

}

if (isMatch === false) {

result.isMatch = false;

return returnObject ? result : false;

}

if (isIgnored(input)) {

if (typeof opts.onIgnore === 'function') {

opts.onIgnore(result);

}

result.isMatch = false;

return returnObject ? result : false;

}

if (typeof opts.onMatch === 'function') {

opts.onMatch(result);

}

return returnObject ? result : true;

};

if (returnState) {

matcher.state = state;

}

return matcher;

};

/\*\*

\* Test `input` with the given `regex`. This is used by the main

\* `picomatch()` function to test the input string.

\*

\* ```js

\* const picomatch = require('picomatch');

\* // picomatch.test(input, regex[, options]);

\*

\* console.log(picomatch.test('foo/bar', /^(?:([^/]\*?)\/([^/]\*?))$/));

\* // { isMatch: true, match: [ 'foo/', 'foo', 'bar' ], output: 'foo/bar' }

\* ```

\* @param {String} `input` String to test.

\* @param {RegExp} `regex`

\* @return {Object} Returns an object with matching info.

\* @api public

\*/

picomatch.test = (input, regex, options, { glob, posix } = {}) => {

if (typeof input !== 'string') {

throw new TypeError('Expected input to be a string');

}

if (input === '') {

return { isMatch: false, output: '' };

}

const opts = options || {};

const format = opts.format || (posix ? utils.toPosixSlashes : null);

let match = input === glob;

let output = (match && format) ? format(input) : input;

if (match === false) {

output = format ? format(input) : input;

match = output === glob;

}

if (match === false || opts.capture === true) {

if (opts.matchBase === true || opts.basename === true) {

match = picomatch.matchBase(input, regex, options, posix);

} else {

match = regex.exec(output);

}

}

return { isMatch: Boolean(match), match, output };

};

/\*\*

\* Match the basename of a filepath.

\*

\* ```js

\* const picomatch = require('picomatch');

\* // picomatch.matchBase(input, glob[, options]);

\* console.log(picomatch.matchBase('foo/bar.js', '\*.js'); // true

\* ```

\* @param {String} `input` String to test.

\* @param {RegExp|String} `glob` Glob pattern or regex created by [.makeRe](#makeRe).

\* @return {Boolean}

\* @api public

\*/

picomatch.matchBase = (input, glob, options) => {

const regex = glob instanceof RegExp ? glob : picomatch.makeRe(glob, options);

return regex.test(utils.basename(input));

};

/\*\*

\* Returns true if \*\*any\*\* of the given glob `patterns` match the specified `string`.

\*

\* ```js

\* const picomatch = require('picomatch');

\* // picomatch.isMatch(string, patterns[, options]);

\*

\* console.log(picomatch.isMatch('a.a', ['b.\*', '\*.a'])); //=> true

\* console.log(picomatch.isMatch('a.a', 'b.\*')); //=> false

\* ```

\* @param {String|Array} str The string to test.

\* @param {String|Array} patterns One or more glob patterns to use for matching.

\* @param {Object} [options] See available [options](#options).

\* @return {Boolean} Returns true if any patterns match `str`

\* @api public

\*/

picomatch.isMatch = (str, patterns, options) => picomatch(patterns, options)(str);

/\*\*

\* Parse a glob pattern to create the source string for a regular

\* expression.

\*

\* ```js

\* const picomatch = require('picomatch');

\* const result = picomatch.parse(pattern[, options]);

\* ```

\* @param {String} `pattern`

\* @param {Object} `options`

\* @return {Object} Returns an object with useful properties and output to be used as a regex source string.

\* @api public

\*/

picomatch.parse = (pattern, options) => {

if (Array.isArray(pattern)) return pattern.map(p => picomatch.parse(p, options));

return parse(pattern, { ...options, fastpaths: false });

};

/\*\*

\* Scan a glob pattern to separate the pattern into segments.

\*

\* ```js

\* const picomatch = require('picomatch');

\* // picomatch.scan(input[, options]);

\*

\* const result = picomatch.scan('!./foo/\*.js');

\* console.log(result);

\* { prefix: '!./',

\* input: '!./foo/\*.js',

\* start: 3,

\* base: 'foo',

\* glob: '\*.js',

\* isBrace: false,

\* isBracket: false,

\* isGlob: true,

\* isExtglob: false,

\* isGlobstar: false,

\* negated: true }

\* ```

\* @param {String} `input` Glob pattern to scan.

\* @param {Object} `options`

\* @return {Object} Returns an object with

\* @api public

\*/

picomatch.scan = (input, options) => scan(input, options);

/\*\*

\* Compile a regular expression from the `state` object returned by the

\* [parse()](#parse) method.

\*

\* @param {Object} `state`

\* @param {Object} `options`

\* @param {Boolean} `returnOutput` Intended for implementors, this argument allows you to return the raw output from the parser.

\* @param {Boolean} `returnState` Adds the state to a `state` property on the returned regex. Useful for implementors and debugging.

\* @return {RegExp}

\* @api public

\*/

picomatch.compileRe = (state, options, returnOutput = false, returnState = false) => {

if (returnOutput === true) {

return state.output;

}

const opts = options || {};

const prepend = opts.contains ? '' : '^';

const append = opts.contains ? '' : '$';

let source = `${prepend}(?:${state.output})${append}`;

if (state && state.negated === true) {

source = `^(?!${source}).\*$`;

}

const regex = picomatch.toRegex(source, options);

if (returnState === true) {

regex.state = state;

}

return regex;

};

/\*\*

\* Create a regular expression from a parsed glob pattern.

\*

\* ```js

\* const picomatch = require('picomatch');

\* const state = picomatch.parse('\*.js');

\* // picomatch.compileRe(state[, options]);

\*

\* console.log(picomatch.compileRe(state));

\* //=> /^(?:(?!\.)(?=.)[^/]\*?\.js)$/

\* ```

\* @param {String} `state` The object returned from the `.parse` method.

\* @param {Object} `options`

\* @param {Boolean} `returnOutput` Implementors may use this argument to return the compiled output, instead of a regular expression. This is not exposed on the options to prevent end-users from mutating the result.

\* @param {Boolean} `returnState` Implementors may use this argument to return the state from the parsed glob with the returned regular expression.

\* @return {RegExp} Returns a regex created from the given pattern.

\* @api public

\*/

picomatch.makeRe = (input, options = {}, returnOutput = false, returnState = false) => {

if (!input || typeof input !== 'string') {

throw new TypeError('Expected a non-empty string');

}

let parsed = { negated: false, fastpaths: true };

if (options.fastpaths !== false && (input[0] === '.' || input[0] === '\*')) {

parsed.output = parse.fastpaths(input, options);

}

if (!parsed.output) {

parsed = parse(input, options);

}

return picomatch.compileRe(parsed, options, returnOutput, returnState);

};

/\*\*

\* Create a regular expression from the given regex source string.

\*

\* ```js

\* const picomatch = require('picomatch');

\* // picomatch.toRegex(source[, options]);

\*

\* const { output } = picomatch.parse('\*.js');

\* console.log(picomatch.toRegex(output));

\* //=> /^(?:(?!\.)(?=.)[^/]\*?\.js)$/

\* ```

\* @param {String} `source` Regular expression source string.

\* @param {Object} `options`

\* @return {RegExp}

\* @api public

\*/

picomatch.toRegex = (source, options) => {

try {

const opts = options || {};

return new RegExp(source, opts.flags || (opts.nocase ? 'i' : ''));

} catch (err) {

if (options && options.debug === true) throw err;

return /$^/;

}

};

/\*\*

\* Picomatch constants.

\* @return {Object}

\*/

picomatch.constants = constants;

/\*\*

\* Expose "picomatch"

\*/

picomatch\_1$1 = picomatch;

return picomatch\_1$1;

}

var picomatch\_1;

var hasRequiredPicomatch;

function requirePicomatch () {

if (hasRequiredPicomatch) return picomatch\_1;

hasRequiredPicomatch = 1;

const pico = /\*@\_\_PURE\_\_\*/ requirePicomatch$1();

const utils = /\*@\_\_PURE\_\_\*/ requireUtils();

function picomatch(glob, options, returnState = false) {

if (options && (options.windows === null || options.windows === undefined)) {

options = { ...options, windows: utils.isWindows() };

}

return pico(glob, options, returnState);

}

Object.assign(picomatch, pico);

picomatch\_1 = picomatch;

return picomatch\_1;

}

var picomatchExports = /\*@\_\_PURE\_\_\*/ requirePicomatch();

var pm = /\*@\_\_PURE\_\_\*/getDefaultExportFromCjs(picomatchExports);

function isArray(arg) {

return Array.isArray(arg);

}

function ensureArray(thing) {

if (isArray(thing))

return thing;

if (thing == null)

return [];

return [thing];

}

const globToTest = (glob) => {

const pattern = glob;

const fn = pm(pattern, { dot: true });

return {

test: (what) => {

const result = fn(what);

return result;

},

};

};

const testTrue = {

test: () => true,

};

const getMatcher = (filter) => {

const bundleTest = "bundle" in filter && filter.bundle != null ? globToTest(filter.bundle) : testTrue;

const fileTest = "file" in filter && filter.file != null ? globToTest(filter.file) : testTrue;

return { bundleTest, fileTest };

};

const createFilter = (include, exclude) => {

const includeMatchers = ensureArray(include).map(getMatcher);

const excludeMatchers = ensureArray(exclude).map(getMatcher);

return (bundleId, id) => {

for (let i = 0; i < excludeMatchers.length; ++i) {

const { bundleTest, fileTest } = excludeMatchers[i];

if (bundleTest.test(bundleId) && fileTest.test(id))

return false;

}

for (let i = 0; i < includeMatchers.length; ++i) {

const { bundleTest, fileTest } = includeMatchers[i];

if (bundleTest.test(bundleId) && fileTest.test(id))

return true;

}

return !includeMatchers.length;

};

};

const throttleFilter = (callback, limit) => {

let waiting = false;

return (val) => {

if (!waiting) {

callback(val);

waiting = true;

setTimeout(() => {

waiting = false;

}, limit);

}

};

};

const prepareFilter = (filt) => {

if (filt === "")

return [];

return (filt

.split(",")

.map((entry) => entry.trim())

.map((entry) => entry.startsWith('"') && entry.endsWith('"') ? entry.substring(1, entry.length - 1) : entry)

.map((entry) => entry.startsWith("'") && entry.endsWith("'") ? entry.substring(1, entry.length - 1) : entry)

.filter((entry) => entry)

.map((entry) => entry.split(":"))

.flatMap((entry) => {

if (entry.length === 0)

return [];

let bundle = null;

let file = null;

if (entry.length === 1 && entry[0]) {

file = entry[0];

return [{ file, bundle }];

}

bundle = entry[0] || null;

file = entry.slice(1).join(":") || null;

return [{ bundle, file }];

}));

};

const useFilter = () => {

const [includeFilter, setIncludeFilter] = h("");

const [excludeFilter, setExcludeFilter] = h("");

const setIncludeFilterTrottled = T(() => throttleFilter(setIncludeFilter, 200), []);

const setExcludeFilterTrottled = T(() => throttleFilter(setExcludeFilter, 200), []);

const isIncluded = T(() => createFilter(prepareFilter(includeFilter), prepareFilter(excludeFilter)), [includeFilter, excludeFilter]);

const getModuleFilterMultiplier = q((bundleId, data) => {

return isIncluded(bundleId, data.id) ? 1 : 0;

}, [isIncluded]);

return {

getModuleFilterMultiplier,

includeFilter,

excludeFilter,

setExcludeFilter: setExcludeFilterTrottled,

setIncludeFilter: setIncludeFilterTrottled,

};

};

function ascending(a, b) {

return a == null || b == null ? NaN : a < b ? -1 : a > b ? 1 : a >= b ? 0 : NaN;

}

function descending(a, b) {

return a == null || b == null ? NaN

: b < a ? -1

: b > a ? 1

: b >= a ? 0

: NaN;

}

function bisector(f) {

let compare1, compare2, delta;

if (f.length !== 2) {

compare1 = ascending;

compare2 = (d, x) => ascending(f(d), x);

delta = (d, x) => f(d) - x;

} else {

compare1 = f === ascending || f === descending ? f : zero$1;

compare2 = f;

delta = f;

}

function left(a, x, lo = 0, hi = a.length) {

if (lo < hi) {

if (compare1(x, x) !== 0) return hi;

do {

const mid = (lo + hi) >>> 1;

if (compare2(a[mid], x) < 0) lo = mid + 1;

else hi = mid;

} while (lo < hi);

}

return lo;

}

function right(a, x, lo = 0, hi = a.length) {

if (lo < hi) {

if (compare1(x, x) !== 0) return hi;

do {

const mid = (lo + hi) >>> 1;

if (compare2(a[mid], x) <= 0) lo = mid + 1;

else hi = mid;

} while (lo < hi);

}

return lo;

}

function center(a, x, lo = 0, hi = a.length) {

const i = left(a, x, lo, hi - 1);

return i > lo && delta(a[i - 1], x) > -delta(a[i], x) ? i - 1 : i;

}

return {left, center, right};

}

function zero$1() {

return 0;

}

function number$1(x) {

return x === null ? NaN : +x;

}

const ascendingBisect = bisector(ascending);

const bisectRight = ascendingBisect.right;

bisector(number$1).center;

class InternMap extends Map {

constructor(entries, key = keyof) {

super();

Object.defineProperties(this, {\_intern: {value: new Map()}, \_key: {value: key}});

if (entries != null) for (const [key, value] of entries) this.set(key, value);

}

get(key) {

return super.get(intern\_get(this, key));

}

has(key) {

return super.has(intern\_get(this, key));

}

set(key, value) {

return super.set(intern\_set(this, key), value);

}

delete(key) {

return super.delete(intern\_delete(this, key));

}

}

function intern\_get({\_intern, \_key}, value) {

const key = \_key(value);

return \_intern.has(key) ? \_intern.get(key) : value;

}

function intern\_set({\_intern, \_key}, value) {

const key = \_key(value);

if (\_intern.has(key)) return \_intern.get(key);

\_intern.set(key, value);

return value;

}

function intern\_delete({\_intern, \_key}, value) {

const key = \_key(value);

if (\_intern.has(key)) {

value = \_intern.get(key);

\_intern.delete(key);

}

return value;

}

function keyof(value) {

return value !== null && typeof value === "object" ? value.valueOf() : value;

}

function identity$2(x) {

return x;

}

function group(values, ...keys) {

return nest(values, identity$2, identity$2, keys);

}

function nest(values, map, reduce, keys) {

return (function regroup(values, i) {

if (i >= keys.length) return reduce(values);

const groups = new InternMap();

const keyof = keys[i++];

let index = -1;

for (const value of values) {

const key = keyof(value, ++index, values);

const group = groups.get(key);

if (group) group.push(value);

else groups.set(key, [value]);

}

for (const [key, values] of groups) {

groups.set(key, regroup(values, i));

}

return map(groups);

})(values, 0);

}

const e10 = Math.sqrt(50),

e5 = Math.sqrt(10),

e2 = Math.sqrt(2);

function tickSpec(start, stop, count) {

const step = (stop - start) / Math.max(0, count),

power = Math.floor(Math.log10(step)),

error = step / Math.pow(10, power),

factor = error >= e10 ? 10 : error >= e5 ? 5 : error >= e2 ? 2 : 1;

let i1, i2, inc;

if (power < 0) {

inc = Math.pow(10, -power) / factor;

i1 = Math.round(start \* inc);

i2 = Math.round(stop \* inc);

if (i1 / inc < start) ++i1;

if (i2 / inc > stop) --i2;

inc = -inc;

} else {

inc = Math.pow(10, power) \* factor;

i1 = Math.round(start / inc);

i2 = Math.round(stop / inc);

if (i1 \* inc < start) ++i1;

if (i2 \* inc > stop) --i2;

}

if (i2 < i1 && 0.5 <= count && count < 2) return tickSpec(start, stop, count \* 2);

return [i1, i2, inc];

}

function ticks(start, stop, count) {

stop = +stop, start = +start, count = +count;

if (!(count > 0)) return [];

if (start === stop) return [start];

const reverse = stop < start, [i1, i2, inc] = reverse ? tickSpec(stop, start, count) : tickSpec(start, stop, count);

if (!(i2 >= i1)) return [];

const n = i2 - i1 + 1, ticks = new Array(n);

if (reverse) {

if (inc < 0) for (let i = 0; i < n; ++i) ticks[i] = (i2 - i) / -inc;

else for (let i = 0; i < n; ++i) ticks[i] = (i2 - i) \* inc;

} else {

if (inc < 0) for (let i = 0; i < n; ++i) ticks[i] = (i1 + i) / -inc;

else for (let i = 0; i < n; ++i) ticks[i] = (i1 + i) \* inc;

}

return ticks;

}

function tickIncrement(start, stop, count) {

stop = +stop, start = +start, count = +count;

return tickSpec(start, stop, count)[2];

}

function tickStep(start, stop, count) {

stop = +stop, start = +start, count = +count;

const reverse = stop < start, inc = reverse ? tickIncrement(stop, start, count) : tickIncrement(start, stop, count);

return (reverse ? -1 : 1) \* (inc < 0 ? 1 / -inc : inc);

}

const TOP\_PADDING = 20;

const PADDING = 2;

const Node = ({ node, onMouseOver, onClick, selected }) => {

const { getModuleColor } = x(StaticContext);

const { backgroundColor, fontColor } = getModuleColor(node);

const { x0, x1, y1, y0, data, children = null } = node;

const textRef = A(null);

const textRectRef = A();

const width = x1 - x0;

const height = y1 - y0;

const textProps = {

"font-size": "0.7em",

"dominant-baseline": "middle",

"text-anchor": "middle",

x: width / 2,

};

if (children != null) {

textProps.y = (TOP\_PADDING + PADDING) / 2;

}

else {

textProps.y = height / 2;

}

\_(() => {

if (width == 0 || height == 0 || !textRef.current) {

return;

}

if (textRectRef.current == null) {

textRectRef.current = textRef.current.getBoundingClientRect();

}

let scale = 1;

if (children != null) {

scale = Math.min((width \* 0.9) / textRectRef.current.width, Math.min(height, TOP\_PADDING + PADDING) / textRectRef.current.height);

scale = Math.min(1, scale);

textRef.current.setAttribute("y", String(Math.min(TOP\_PADDING + PADDING, height) / 2 / scale));

textRef.current.setAttribute("x", String(width / 2 / scale));

}

else {

scale = Math.min((width \* 0.9) / textRectRef.current.width, (height \* 0.9) / textRectRef.current.height);

scale = Math.min(1, scale);

textRef.current.setAttribute("y", String(height / 2 / scale));

textRef.current.setAttribute("x", String(width / 2 / scale));

}

textRef.current.setAttribute("transform", `scale(${scale.toFixed(2)})`);

}, [children, height, width]);

if (width == 0 || height == 0) {

return null;

}

return (u$1("g", { className: "node", transform: `translate(${x0},${y0})`, onClick: (event) => {

event.stopPropagation();

onClick(node);

}, onMouseOver: (event) => {

event.stopPropagation();

onMouseOver(node);

}, children: [u$1("rect", { fill: backgroundColor, rx: 2, ry: 2, width: x1 - x0, height: y1 - y0, stroke: selected ? "#fff" : undefined, "stroke-width": selected ? 2 : undefined }), u$1("text", Object.assign({ ref: textRef, fill: fontColor, onClick: (event) => {

var \_a;

if (((\_a = window.getSelection()) === null || \_a === void 0 ? void 0 : \_a.toString()) !== "") {

event.stopPropagation();

}

} }, textProps, { children: data.name }))] }));

};

const TreeMap = ({ root, onNodeHover, selectedNode, onNodeClick, }) => {

const { width, height, getModuleIds } = x(StaticContext);

console.time("layering");

const nestedData = T(() => {

const nestedDataMap = group(root.descendants(), (d) => d.height);

const nestedData = Array.from(nestedDataMap, ([key, values]) => ({

key,

values,

}));

nestedData.sort((a, b) => b.key - a.key);

return nestedData;

}, [root]);

console.timeEnd("layering");

return (u$1("svg", { xmlns: "http://www.w3.org/2000/svg", viewBox: `0 0 ${width} ${height}`, children: nestedData.map(({ key, values }) => {

return (u$1("g", { className: "layer", children: values.map((node) => {

return (u$1(Node, { node: node, onMouseOver: onNodeHover, selected: selectedNode === node, onClick: onNodeClick }, getModuleIds(node.data).nodeUid.id));

}) }, key));

}) }));

};

var bytes = {exports: {}};

/\*!

\* bytes

\* Copyright(c) 2012-2014 TJ Holowaychuk

\* Copyright(c) 2015 Jed Watson

\* MIT Licensed

\*/

var hasRequiredBytes;

function requireBytes () {

if (hasRequiredBytes) return bytes.exports;

hasRequiredBytes = 1;

/\*\*

\* Module exports.

\* @public

\*/

bytes.exports = bytes$1;

bytes.exports.format = format;

bytes.exports.parse = parse;

/\*\*

\* Module variables.

\* @private

\*/

var formatThousandsRegExp = /\B(?=(\d{3})+(?!\d))/g;

var formatDecimalsRegExp = /(?:\.0\*|(\.[^0]+)0+)$/;

var map = {

b: 1,

kb: 1 << 10,

mb: 1 << 20,

gb: 1 << 30,

tb: Math.pow(1024, 4),

pb: Math.pow(1024, 5),

};

var parseRegExp = /^((-|\+)?(\d+(?:\.\d+)?)) \*(kb|mb|gb|tb|pb)$/i;

/\*\*

\* Convert the given value in bytes into a string or parse to string to an integer in bytes.

\*

\* @param {string|number} value

\* @param {{

\* case: [string],

\* decimalPlaces: [number]

\* fixedDecimals: [boolean]

\* thousandsSeparator: [string]

\* unitSeparator: [string]

\* }} [options] bytes options.

\*

\* @returns {string|number|null}

\*/

function bytes$1(value, options) {

if (typeof value === 'string') {

return parse(value);

}

if (typeof value === 'number') {

return format(value, options);

}

return null;

}

/\*\*

\* Format the given value in bytes into a string.

\*

\* If the value is negative, it is kept as such. If it is a float,

\* it is rounded.

\*

\* @param {number} value

\* @param {object} [options]

\* @param {number} [options.decimalPlaces=2]

\* @param {number} [options.fixedDecimals=false]

\* @param {string} [options.thousandsSeparator=]

\* @param {string} [options.unit=]

\* @param {string} [options.unitSeparator=]

\*

\* @returns {string|null}

\* @public

\*/

function format(value, options) {

if (!Number.isFinite(value)) {

return null;

}

var mag = Math.abs(value);

var thousandsSeparator = (options && options.thousandsSeparator) || '';

var unitSeparator = (options && options.unitSeparator) || '';

var decimalPlaces = (options && options.decimalPlaces !== undefined) ? options.decimalPlaces : 2;

var fixedDecimals = Boolean(options && options.fixedDecimals);

var unit = (options && options.unit) || '';

if (!unit || !map[unit.toLowerCase()]) {

if (mag >= map.pb) {

unit = 'PB';

} else if (mag >= map.tb) {

unit = 'TB';

} else if (mag >= map.gb) {

unit = 'GB';

} else if (mag >= map.mb) {

unit = 'MB';

} else if (mag >= map.kb) {

unit = 'KB';

} else {

unit = 'B';

}

}

var val = value / map[unit.toLowerCase()];

var str = val.toFixed(decimalPlaces);

if (!fixedDecimals) {

str = str.replace(formatDecimalsRegExp, '$1');

}

if (thousandsSeparator) {

str = str.split('.').map(function (s, i) {

return i === 0

? s.replace(formatThousandsRegExp, thousandsSeparator)

: s

}).join('.');

}

return str + unitSeparator + unit;

}

/\*\*

\* Parse the string value into an integer in bytes.

\*

\* If no unit is given, it is assumed the value is in bytes.

\*

\* @param {number|string} val

\*

\* @returns {number|null}

\* @public

\*/

function parse(val) {

if (typeof val === 'number' && !isNaN(val)) {

return val;

}

if (typeof val !== 'string') {

return null;

}

var results = parseRegExp.exec(val);

var floatValue;

var unit = 'b';

if (!results) {

floatValue = parseInt(val, 10);

unit = 'b';

} else {

floatValue = parseFloat(results[1]);

unit = results[4].toLowerCase();

}

if (isNaN(floatValue)) {

return null;

}

return Math.floor(map[unit] \* floatValue);

}

return bytes.exports;

}

var bytesExports = requireBytes();

const Tooltip\_marginX = 10;

const Tooltip\_marginY = 30;

const SOURCEMAP\_RENDERED = (u$1("span", { children: [" ", u$1("b", { children: LABELS.renderedLength }), " is a number of characters in the file after individual and ", u$1("br", {}), " ", "whole bundle transformations according to sourcemap."] }));

const RENDRED = (u$1("span", { children: [u$1("b", { children: LABELS.renderedLength }), " is a byte size of individual file after transformations and treeshake."] }));

const COMPRESSED = (u$1("span", { children: [u$1("b", { children: LABELS.gzipLength }), " and ", u$1("b", { children: LABELS.brotliLength }), " is a byte size of individual file after individual transformations,", u$1("br", {}), " treeshake and compression."] }));

const Tooltip = ({ node, visible, root, sizeProperty, }) => {

const { availableSizeProperties, getModuleSize, data } = x(StaticContext);

const ref = A(null);

const [style, setStyle] = h({});

const content = T(() => {

if (!node)

return null;

const mainSize = getModuleSize(node.data, sizeProperty);

const percentageNum = (100 \* mainSize) / getModuleSize(root.data, sizeProperty);

const percentage = percentageNum.toFixed(2);

const percentageString = percentage + "%";

const path = node

.ancestors()

.reverse()

.map((d) => d.data.name)

.join("/");

let dataNode = null;

if (!isModuleTree(node.data)) {

const mainUid = data.nodeParts[node.data.uid].metaUid;

dataNode = data.nodeMetas[mainUid];

}

return (u$1(k$1, { children: [u$1("div", { children: path }), availableSizeProperties.map((sizeProp) => {

if (sizeProp === sizeProperty) {

return (u$1("div", { children: [u$1("b", { children: [LABELS[sizeProp], ": ", bytesExports.format(mainSize)] }), " ", "(", percentageString, ")"] }, sizeProp));

}

else {

return (u$1("div", { children: [LABELS[sizeProp], ": ", bytesExports.format(getModuleSize(node.data, sizeProp))] }, sizeProp));

}

}), u$1("br", {}), dataNode && dataNode.importedBy.length > 0 && (u$1("div", { children: [u$1("div", { children: [u$1("b", { children: "Imported By" }), ":"] }), dataNode.importedBy.map(({ uid }) => {

const id = data.nodeMetas[uid].id;

return u$1("div", { children: id }, id);

})] })), u$1("br", {}), u$1("small", { children: data.options.sourcemap ? SOURCEMAP\_RENDERED : RENDRED }), (data.options.gzip || data.options.brotli) && (u$1(k$1, { children: [u$1("br", {}), u$1("small", { children: COMPRESSED })] }))] }));

}, [availableSizeProperties, data, getModuleSize, node, root.data, sizeProperty]);

const updatePosition = (mouseCoords) => {

if (!ref.current)

return;

const pos = {

left: mouseCoords.x + Tooltip\_marginX,

top: mouseCoords.y + Tooltip\_marginY,

};

const boundingRect = ref.current.getBoundingClientRect();

if (pos.left + boundingRect.width > window.innerWidth) {

pos.left = Math.max(0, window.innerWidth - boundingRect.width);

}

if (pos.top + boundingRect.height > window.innerHeight) {

pos.top = Math.max(0, mouseCoords.y - Tooltip\_marginY - boundingRect.height);

}

setStyle(pos);

};

y(() => {

const handleMouseMove = (event) => {

updatePosition({

x: event.pageX,

y: event.pageY,

});

};

document.addEventListener("mousemove", handleMouseMove, true);

return () => {

document.removeEventListener("mousemove", handleMouseMove, true);

};

}, []);

return (u$1("div", { className: `tooltip ${visible ? "" : "tooltip-hidden"}`, ref: ref, style: style, children: content }));

};

const Chart = ({ root, sizeProperty, selectedNode, setSelectedNode, }) => {

const [showTooltip, setShowTooltip] = h(false);

const [tooltipNode, setTooltipNode] = h(undefined);

y(() => {

const handleMouseOut = () => {

setShowTooltip(false);

};

document.addEventListener("mouseover", handleMouseOut);

return () => {

document.removeEventListener("mouseover", handleMouseOut);

};

}, []);

return (u$1(k$1, { children: [u$1(TreeMap, { root: root, onNodeHover: (node) => {

setTooltipNode(node);

setShowTooltip(true);

}, selectedNode: selectedNode, onNodeClick: (node) => {

setSelectedNode(selectedNode === node ? undefined : node);

} }), u$1(Tooltip, { visible: showTooltip, node: tooltipNode, root: root, sizeProperty: sizeProperty })] }));

};

const Main = () => {

const { availableSizeProperties, rawHierarchy, getModuleSize, layout, data } = x(StaticContext);

const [sizeProperty, setSizeProperty] = h(availableSizeProperties[0]);

const [selectedNode, setSelectedNode] = h(undefined);

const { getModuleFilterMultiplier, setExcludeFilter, setIncludeFilter } = useFilter();

console.time("getNodeSizeMultiplier");

const getNodeSizeMultiplier = T(() => {

const selectedMultiplier = 1; // selectedSize < rootSize \* increaseFactor ? (rootSize \* increaseFactor) / selectedSize : rootSize / selectedSize;

const nonSelectedMultiplier = 0; // 1 / selectedMultiplier

if (selectedNode === undefined) {

return () => 1;

}

else if (isModuleTree(selectedNode.data)) {

const leaves = new Set(selectedNode.leaves().map((d) => d.data));

return (node) => {

if (leaves.has(node)) {

return selectedMultiplier;

}

return nonSelectedMultiplier;

};

}

else {

return (node) => {

if (node === selectedNode.data) {

return selectedMultiplier;

}

return nonSelectedMultiplier;

};

}

}, [getModuleSize, rawHierarchy.data, selectedNode, sizeProperty]);

console.timeEnd("getNodeSizeMultiplier");

console.time("root hierarchy compute");

const root = T(() => {

const rootWithSizesAndSorted = rawHierarchy

.sum((node) => {

var \_a;

if (isModuleTree(node))

return 0;

const meta = data.nodeMetas[data.nodeParts[node.uid].metaUid];

/\* eslint-disable typescript/no-non-null-asserted-optional-chain typescript/no-extra-non-null-assertion \*/

const bundleId = (\_a = Object.entries(meta.moduleParts).find(([, uid]) => uid == node.uid)) === null || \_a === void 0 ? void 0 : \_a[0];

const ownSize = getModuleSize(node, sizeProperty);

const zoomMultiplier = getNodeSizeMultiplier(node);

const filterMultiplier = getModuleFilterMultiplier(bundleId, meta);

return ownSize \* zoomMultiplier \* filterMultiplier;

})

.sort((a, b) => getModuleSize(a.data, sizeProperty) - getModuleSize(b.data, sizeProperty));

return layout(rootWithSizesAndSorted);

}, [

data,

getModuleFilterMultiplier,

getModuleSize,

getNodeSizeMultiplier,

layout,

rawHierarchy,

sizeProperty,

]);

console.timeEnd("root hierarchy compute");

return (u$1(k$1, { children: [u$1(SideBar, { sizeProperty: sizeProperty, availableSizeProperties: availableSizeProperties, setSizeProperty: setSizeProperty, onExcludeChange: setExcludeFilter, onIncludeChange: setIncludeFilter }), u$1(Chart, { root: root, sizeProperty: sizeProperty, selectedNode: selectedNode, setSelectedNode: setSelectedNode })] }));

};

function initRange(domain, range) {

switch (arguments.length) {

case 0: break;

case 1: this.range(domain); break;

default: this.range(range).domain(domain); break;

}

return this;

}

function initInterpolator(domain, interpolator) {

switch (arguments.length) {

case 0: break;

case 1: {

if (typeof domain === "function") this.interpolator(domain);

else this.range(domain);

break;

}

default: {

this.domain(domain);

if (typeof interpolator === "function") this.interpolator(interpolator);

else this.range(interpolator);

break;

}

}

return this;

}

function define(constructor, factory, prototype) {

constructor.prototype = factory.prototype = prototype;

prototype.constructor = constructor;

}

function extend(parent, definition) {

var prototype = Object.create(parent.prototype);

for (var key in definition) prototype[key] = definition[key];

return prototype;

}

function Color() {}

var darker = 0.7;

var brighter = 1 / darker;

var reI = "\\s\*([+-]?\\d+)\\s\*",

reN = "\\s\*([+-]?(?:\\d\*\\.)?\\d+(?:[eE][+-]?\\d+)?)\\s\*",

reP = "\\s\*([+-]?(?:\\d\*\\.)?\\d+(?:[eE][+-]?\\d+)?)%\\s\*",

reHex = /^#([0-9a-f]{3,8})$/,

reRgbInteger = new RegExp(`^rgb\\(${reI},${reI},${reI}\\)$`),

reRgbPercent = new RegExp(`^rgb\\(${reP},${reP},${reP}\\)$`),

reRgbaInteger = new RegExp(`^rgba\\(${reI},${reI},${reI},${reN}\\)$`),

reRgbaPercent = new RegExp(`^rgba\\(${reP},${reP},${reP},${reN}\\)$`),

reHslPercent = new RegExp(`^hsl\\(${reN},${reP},${reP}\\)$`),

reHslaPercent = new RegExp(`^hsla\\(${reN},${reP},${reP},${reN}\\)$`);

var named = {

aliceblue: 0xf0f8ff,

antiquewhite: 0xfaebd7,

aqua: 0x00ffff,

aquamarine: 0x7fffd4,

azure: 0xf0ffff,

beige: 0xf5f5dc,

bisque: 0xffe4c4,

black: 0x000000,

blanchedalmond: 0xffebcd,

blue: 0x0000ff,

blueviolet: 0x8a2be2,

brown: 0xa52a2a,

burlywood: 0xdeb887,

cadetblue: 0x5f9ea0,

chartreuse: 0x7fff00,

chocolate: 0xd2691e,

coral: 0xff7f50,

cornflowerblue: 0x6495ed,

cornsilk: 0xfff8dc,

crimson: 0xdc143c,

cyan: 0x00ffff,

darkblue: 0x00008b,

darkcyan: 0x008b8b,

darkgoldenrod: 0xb8860b,

darkgray: 0xa9a9a9,

darkgreen: 0x006400,

darkgrey: 0xa9a9a9,

darkkhaki: 0xbdb76b,

darkmagenta: 0x8b008b,

darkolivegreen: 0x556b2f,

darkorange: 0xff8c00,

darkorchid: 0x9932cc,

darkred: 0x8b0000,

darksalmon: 0xe9967a,

darkseagreen: 0x8fbc8f,

darkslateblue: 0x483d8b,

darkslategray: 0x2f4f4f,

darkslategrey: 0x2f4f4f,

darkturquoise: 0x00ced1,

darkviolet: 0x9400d3,

deeppink: 0xff1493,

deepskyblue: 0x00bfff,

dimgray: 0x696969,

dimgrey: 0x696969,

dodgerblue: 0x1e90ff,

firebrick: 0xb22222,

floralwhite: 0xfffaf0,

forestgreen: 0x228b22,

fuchsia: 0xff00ff,

gainsboro: 0xdcdcdc,

ghostwhite: 0xf8f8ff,

gold: 0xffd700,

goldenrod: 0xdaa520,

gray: 0x808080,

green: 0x008000,

greenyellow: 0xadff2f,

grey: 0x808080,

honeydew: 0xf0fff0,

hotpink: 0xff69b4,

indianred: 0xcd5c5c,

indigo: 0x4b0082,

ivory: 0xfffff0,

khaki: 0xf0e68c,

lavender: 0xe6e6fa,

lavenderblush: 0xfff0f5,

lawngreen: 0x7cfc00,

lemonchiffon: 0xfffacd,

lightblue: 0xadd8e6,

lightcoral: 0xf08080,

lightcyan: 0xe0ffff,

lightgoldenrodyellow: 0xfafad2,

lightgray: 0xd3d3d3,

lightgreen: 0x90ee90,

lightgrey: 0xd3d3d3,

lightpink: 0xffb6c1,

lightsalmon: 0xffa07a,

lightseagreen: 0x20b2aa,

lightskyblue: 0x87cefa,

lightslategray: 0x778899,

lightslategrey: 0x778899,

lightsteelblue: 0xb0c4de,

lightyellow: 0xffffe0,

lime: 0x00ff00,

limegreen: 0x32cd32,

linen: 0xfaf0e6,

magenta: 0xff00ff,

maroon: 0x800000,

mediumaquamarine: 0x66cdaa,

mediumblue: 0x0000cd,

mediumorchid: 0xba55d3,

mediumpurple: 0x9370db,

mediumseagreen: 0x3cb371,

mediumslateblue: 0x7b68ee,

mediumspringgreen: 0x00fa9a,

mediumturquoise: 0x48d1cc,

mediumvioletred: 0xc71585,

midnightblue: 0x191970,

mintcream: 0xf5fffa,

mistyrose: 0xffe4e1,

moccasin: 0xffe4b5,

navajowhite: 0xffdead,

navy: 0x000080,

oldlace: 0xfdf5e6,

olive: 0x808000,

olivedrab: 0x6b8e23,

orange: 0xffa500,

orangered: 0xff4500,

orchid: 0xda70d6,

palegoldenrod: 0xeee8aa,

palegreen: 0x98fb98,

paleturquoise: 0xafeeee,

palevioletred: 0xdb7093,

papayawhip: 0xffefd5,

peachpuff: 0xffdab9,

peru: 0xcd853f,

pink: 0xffc0cb,

plum: 0xdda0dd,

powderblue: 0xb0e0e6,

purple: 0x800080,

rebeccapurple: 0x663399,

red: 0xff0000,

rosybrown: 0xbc8f8f,

royalblue: 0x4169e1,

saddlebrown: 0x8b4513,

salmon: 0xfa8072,

sandybrown: 0xf4a460,

seagreen: 0x2e8b57,

seashell: 0xfff5ee,

sienna: 0xa0522d,

silver: 0xc0c0c0,

skyblue: 0x87ceeb,

slateblue: 0x6a5acd,

slategray: 0x708090,

slategrey: 0x708090,

snow: 0xfffafa,

springgreen: 0x00ff7f,

steelblue: 0x4682b4,

tan: 0xd2b48c,

teal: 0x008080,

thistle: 0xd8bfd8,

tomato: 0xff6347,

turquoise: 0x40e0d0,

violet: 0xee82ee,

wheat: 0xf5deb3,

white: 0xffffff,

whitesmoke: 0xf5f5f5,

yellow: 0xffff00,

yellowgreen: 0x9acd32

};

define(Color, color, {

copy(channels) {

return Object.assign(new this.constructor, this, channels);

},

displayable() {

return this.rgb().displayable();

},

hex: color\_formatHex, // Deprecated! Use color.formatHex.

formatHex: color\_formatHex,

formatHex8: color\_formatHex8,

formatHsl: color\_formatHsl,

formatRgb: color\_formatRgb,

toString: color\_formatRgb

});

function color\_formatHex() {

return this.rgb().formatHex();

}

function color\_formatHex8() {

return this.rgb().formatHex8();

}

function color\_formatHsl() {

return hslConvert(this).formatHsl();

}

function color\_formatRgb() {

return this.rgb().formatRgb();

}

function color(format) {

var m, l;

format = (format + "").trim().toLowerCase();

return (m = reHex.exec(format)) ? (l = m[1].length, m = parseInt(m[1], 16), l === 6 ? rgbn(m) // #ff0000

: l === 3 ? new Rgb((m >> 8 & 0xf) | (m >> 4 & 0xf0), (m >> 4 & 0xf) | (m & 0xf0), ((m & 0xf) << 4) | (m & 0xf), 1) // #f00

: l === 8 ? rgba(m >> 24 & 0xff, m >> 16 & 0xff, m >> 8 & 0xff, (m & 0xff) / 0xff) // #ff000000

: l === 4 ? rgba((m >> 12 & 0xf) | (m >> 8 & 0xf0), (m >> 8 & 0xf) | (m >> 4 & 0xf0), (m >> 4 & 0xf) | (m & 0xf0), (((m & 0xf) << 4) | (m & 0xf)) / 0xff) // #f000

: null) // invalid hex

: (m = reRgbInteger.exec(format)) ? new Rgb(m[1], m[2], m[3], 1) // rgb(255, 0, 0)

: (m = reRgbPercent.exec(format)) ? new Rgb(m[1] \* 255 / 100, m[2] \* 255 / 100, m[3] \* 255 / 100, 1) // rgb(100%, 0%, 0%)

: (m = reRgbaInteger.exec(format)) ? rgba(m[1], m[2], m[3], m[4]) // rgba(255, 0, 0, 1)

: (m = reRgbaPercent.exec(format)) ? rgba(m[1] \* 255 / 100, m[2] \* 255 / 100, m[3] \* 255 / 100, m[4]) // rgb(100%, 0%, 0%, 1)

: (m = reHslPercent.exec(format)) ? hsla(m[1], m[2] / 100, m[3] / 100, 1) // hsl(120, 50%, 50%)

: (m = reHslaPercent.exec(format)) ? hsla(m[1], m[2] / 100, m[3] / 100, m[4]) // hsla(120, 50%, 50%, 1)

: named.hasOwnProperty(format) ? rgbn(named[format]) // eslint-disable-line no-prototype-builtins

: format === "transparent" ? new Rgb(NaN, NaN, NaN, 0)

: null;

}

function rgbn(n) {

return new Rgb(n >> 16 & 0xff, n >> 8 & 0xff, n & 0xff, 1);

}

function rgba(r, g, b, a) {

if (a <= 0) r = g = b = NaN;

return new Rgb(r, g, b, a);

}

function rgbConvert(o) {

if (!(o instanceof Color)) o = color(o);

if (!o) return new Rgb;

o = o.rgb();

return new Rgb(o.r, o.g, o.b, o.opacity);

}

function rgb$1(r, g, b, opacity) {

return arguments.length === 1 ? rgbConvert(r) : new Rgb(r, g, b, opacity == null ? 1 : opacity);

}

function Rgb(r, g, b, opacity) {

this.r = +r;

this.g = +g;

this.b = +b;

this.opacity = +opacity;

}

define(Rgb, rgb$1, extend(Color, {

brighter(k) {

k = k == null ? brighter : Math.pow(brighter, k);

return new Rgb(this.r \* k, this.g \* k, this.b \* k, this.opacity);

},

darker(k) {

k = k == null ? darker : Math.pow(darker, k);

return new Rgb(this.r \* k, this.g \* k, this.b \* k, this.opacity);

},

rgb() {

return this;

},

clamp() {

return new Rgb(clampi(this.r), clampi(this.g), clampi(this.b), clampa(this.opacity));

},

displayable() {

return (-0.5 <= this.r && this.r < 255.5)

&& (-0.5 <= this.g && this.g < 255.5)

&& (-0.5 <= this.b && this.b < 255.5)

&& (0 <= this.opacity && this.opacity <= 1);

},

hex: rgb\_formatHex, // Deprecated! Use color.formatHex.

formatHex: rgb\_formatHex,

formatHex8: rgb\_formatHex8,

formatRgb: rgb\_formatRgb,

toString: rgb\_formatRgb

}));

function rgb\_formatHex() {

return `#${hex(this.r)}${hex(this.g)}${hex(this.b)}`;

}

function rgb\_formatHex8() {

return `#${hex(this.r)}${hex(this.g)}${hex(this.b)}${hex((isNaN(this.opacity) ? 1 : this.opacity) \* 255)}`;

}

function rgb\_formatRgb() {

const a = clampa(this.opacity);

return `${a === 1 ? "rgb(" : "rgba("}${clampi(this.r)}, ${clampi(this.g)}, ${clampi(this.b)}${a === 1 ? ")" : `, ${a})`}`;

}

function clampa(opacity) {

return isNaN(opacity) ? 1 : Math.max(0, Math.min(1, opacity));

}

function clampi(value) {

return Math.max(0, Math.min(255, Math.round(value) || 0));

}

function hex(value) {

value = clampi(value);

return (value < 16 ? "0" : "") + value.toString(16);

}

function hsla(h, s, l, a) {

if (a <= 0) h = s = l = NaN;

else if (l <= 0 || l >= 1) h = s = NaN;

else if (s <= 0) h = NaN;

return new Hsl(h, s, l, a);

}

function hslConvert(o) {

if (o instanceof Hsl) return new Hsl(o.h, o.s, o.l, o.opacity);

if (!(o instanceof Color)) o = color(o);

if (!o) return new Hsl;

if (o instanceof Hsl) return o;

o = o.rgb();

var r = o.r / 255,

g = o.g / 255,

b = o.b / 255,

min = Math.min(r, g, b),

max = Math.max(r, g, b),

h = NaN,

s = max - min,

l = (max + min) / 2;

if (s) {

if (r === max) h = (g - b) / s + (g < b) \* 6;

else if (g === max) h = (b - r) / s + 2;

else h = (r - g) / s + 4;

s /= l < 0.5 ? max + min : 2 - max - min;

h \*= 60;

} else {

s = l > 0 && l < 1 ? 0 : h;

}

return new Hsl(h, s, l, o.opacity);

}

function hsl(h, s, l, opacity) {

return arguments.length === 1 ? hslConvert(h) : new Hsl(h, s, l, opacity == null ? 1 : opacity);

}

function Hsl(h, s, l, opacity) {

this.h = +h;

this.s = +s;

this.l = +l;

this.opacity = +opacity;

}

define(Hsl, hsl, extend(Color, {

brighter(k) {

k = k == null ? brighter : Math.pow(brighter, k);

return new Hsl(this.h, this.s, this.l \* k, this.opacity);

},

darker(k) {

k = k == null ? darker : Math.pow(darker, k);

return new Hsl(this.h, this.s, this.l \* k, this.opacity);

},

rgb() {

var h = this.h % 360 + (this.h < 0) \* 360,

s = isNaN(h) || isNaN(this.s) ? 0 : this.s,

l = this.l,

m2 = l + (l < 0.5 ? l : 1 - l) \* s,

m1 = 2 \* l - m2;

return new Rgb(

hsl2rgb(h >= 240 ? h - 240 : h + 120, m1, m2),

hsl2rgb(h, m1, m2),

hsl2rgb(h < 120 ? h + 240 : h - 120, m1, m2),

this.opacity

);

},

clamp() {

return new Hsl(clamph(this.h), clampt(this.s), clampt(this.l), clampa(this.opacity));

},

displayable() {

return (0 <= this.s && this.s <= 1 || isNaN(this.s))

&& (0 <= this.l && this.l <= 1)

&& (0 <= this.opacity && this.opacity <= 1);

},

formatHsl() {

const a = clampa(this.opacity);

return `${a === 1 ? "hsl(" : "hsla("}${clamph(this.h)}, ${clampt(this.s) \* 100}%, ${clampt(this.l) \* 100}%${a === 1 ? ")" : `, ${a})`}`;

}

}));

function clamph(value) {

value = (value || 0) % 360;

return value < 0 ? value + 360 : value;

}

function clampt(value) {

return Math.max(0, Math.min(1, value || 0));

}

/\* From FvD 13.37, CSS Color Module Level 3 \*/

function hsl2rgb(h, m1, m2) {

return (h < 60 ? m1 + (m2 - m1) \* h / 60

: h < 180 ? m2

: h < 240 ? m1 + (m2 - m1) \* (240 - h) / 60

: m1) \* 255;

}

var constant = x => () => x;

function linear$1(a, d) {

return function(t) {

return a + t \* d;

};

}

function exponential(a, b, y) {

return a = Math.pow(a, y), b = Math.pow(b, y) - a, y = 1 / y, function(t) {

return Math.pow(a + t \* b, y);

};

}

function gamma(y) {

return (y = +y) === 1 ? nogamma : function(a, b) {

return b - a ? exponential(a, b, y) : constant(isNaN(a) ? b : a);

};

}

function nogamma(a, b) {

var d = b - a;

return d ? linear$1(a, d) : constant(isNaN(a) ? b : a);

}

var rgb = (function rgbGamma(y) {

var color = gamma(y);

function rgb(start, end) {

var r = color((start = rgb$1(start)).r, (end = rgb$1(end)).r),

g = color(start.g, end.g),

b = color(start.b, end.b),

opacity = nogamma(start.opacity, end.opacity);

return function(t) {

start.r = r(t);

start.g = g(t);

start.b = b(t);

start.opacity = opacity(t);

return start + "";

};

}

rgb.gamma = rgbGamma;

return rgb;

})(1);

function numberArray(a, b) {

if (!b) b = [];

var n = a ? Math.min(b.length, a.length) : 0,

c = b.slice(),

i;

return function(t) {

for (i = 0; i < n; ++i) c[i] = a[i] \* (1 - t) + b[i] \* t;

return c;

};

}

function isNumberArray(x) {

return ArrayBuffer.isView(x) && !(x instanceof DataView);

}

function genericArray(a, b) {

var nb = b ? b.length : 0,

na = a ? Math.min(nb, a.length) : 0,

x = new Array(na),

c = new Array(nb),

i;

for (i = 0; i < na; ++i) x[i] = interpolate(a[i], b[i]);

for (; i < nb; ++i) c[i] = b[i];

return function(t) {

for (i = 0; i < na; ++i) c[i] = x[i](t);

return c;

};

}

function date(a, b) {

var d = new Date;

return a = +a, b = +b, function(t) {

return d.setTime(a \* (1 - t) + b \* t), d;

};

}

function interpolateNumber(a, b) {

return a = +a, b = +b, function(t) {

return a \* (1 - t) + b \* t;

};

}

function object(a, b) {

var i = {},

c = {},

k;

if (a === null || typeof a !== "object") a = {};

if (b === null || typeof b !== "object") b = {};

for (k in b) {

if (k in a) {

i[k] = interpolate(a[k], b[k]);

} else {

c[k] = b[k];

}

}

return function(t) {

for (k in i) c[k] = i[k](t);

return c;

};

}

var reA = /[-+]?(?:\d+\.?\d\*|\.?\d+)(?:[eE][-+]?\d+)?/g,

reB = new RegExp(reA.source, "g");

function zero(b) {

return function() {

return b;

};

}

function one(b) {

return function(t) {

return b(t) + "";

};

}

function string(a, b) {

var bi = reA.lastIndex = reB.lastIndex = 0, // scan index for next number in b

am, // current match in a

bm, // current match in b

bs, // string preceding current number in b, if any

i = -1, // index in s

s = [], // string constants and placeholders

q = []; // number interpolators

a = a + "", b = b + "";

while ((am = reA.exec(a))

&& (bm = reB.exec(b))) {

if ((bs = bm.index) > bi) { // a string precedes the next number in b

bs = b.slice(bi, bs);

if (s[i]) s[i] += bs; // coalesce with previous string

else s[++i] = bs;

}

if ((am = am[0]) === (bm = bm[0])) { // numbers in a & b match

if (s[i]) s[i] += bm; // coalesce with previous string

else s[++i] = bm;

} else { // interpolate non-matching numbers

s[++i] = null;

q.push({i: i, x: interpolateNumber(am, bm)});

}

bi = reB.lastIndex;

}

if (bi < b.length) {

bs = b.slice(bi);

if (s[i]) s[i] += bs; // coalesce with previous string

else s[++i] = bs;

}

return s.length < 2 ? (q[0]

? one(q[0].x)

: zero(b))

: (b = q.length, function(t) {

for (var i = 0, o; i < b; ++i) s[(o = q[i]).i] = o.x(t);

return s.join("");

});

}

function interpolate(a, b) {

var t = typeof b, c;

return b == null || t === "boolean" ? constant(b)

: (t === "number" ? interpolateNumber

: t === "string" ? ((c = color(b)) ? (b = c, rgb) : string)

: b instanceof color ? rgb

: b instanceof Date ? date

: isNumberArray(b) ? numberArray

: Array.isArray(b) ? genericArray

: typeof b.valueOf !== "function" && typeof b.toString !== "function" || isNaN(b) ? object

: interpolateNumber)(a, b);

}

function interpolateRound(a, b) {

return a = +a, b = +b, function(t) {

return Math.round(a \* (1 - t) + b \* t);

};

}

function constants(x) {

return function() {

return x;

};

}

function number(x) {

return +x;

}

var unit = [0, 1];

function identity$1(x) {

return x;

}

function normalize(a, b) {

return (b -= (a = +a))

? function(x) { return (x - a) / b; }

: constants(isNaN(b) ? NaN : 0.5);

}

function clamper(a, b) {

var t;

if (a > b) t = a, a = b, b = t;

return function(x) { return Math.max(a, Math.min(b, x)); };

}

function bimap(domain, range, interpolate) {

var d0 = domain[0], d1 = domain[1], r0 = range[0], r1 = range[1];

if (d1 < d0) d0 = normalize(d1, d0), r0 = interpolate(r1, r0);

else d0 = normalize(d0, d1), r0 = interpolate(r0, r1);

return function(x) { return r0(d0(x)); };

}

function polymap(domain, range, interpolate) {

var j = Math.min(domain.length, range.length) - 1,

d = new Array(j),

r = new Array(j),

i = -1;

if (domain[j] < domain[0]) {

domain = domain.slice().reverse();

range = range.slice().reverse();

}

while (++i < j) {

d[i] = normalize(domain[i], domain[i + 1]);

r[i] = interpolate(range[i], range[i + 1]);

}

return function(x) {

var i = bisectRight(domain, x, 1, j) - 1;

return r[i](d[i](x));

};

}

function copy$1(source, target) {

return target

.domain(source.domain())

.range(source.range())

.interpolate(source.interpolate())

.clamp(source.clamp())

.unknown(source.unknown());

}

function transformer$1() {

var domain = unit,

range = unit,

interpolate$1 = interpolate,

transform,

untransform,

unknown,

clamp = identity$1,

piecewise,

output,

input;

function rescale() {

var n = Math.min(domain.length, range.length);

if (clamp !== identity$1) clamp = clamper(domain[0], domain[n - 1]);

piecewise = n > 2 ? polymap : bimap;

output = input = null;

return scale;

}

function scale(x) {

return x == null || isNaN(x = +x) ? unknown : (output || (output = piecewise(domain.map(transform), range, interpolate$1)))(transform(clamp(x)));

}

scale.invert = function(y) {

return clamp(untransform((input || (input = piecewise(range, domain.map(transform), interpolateNumber)))(y)));

};

scale.domain = function(\_) {

return arguments.length ? (domain = Array.from(\_, number), rescale()) : domain.slice();

};

scale.range = function(\_) {

return arguments.length ? (range = Array.from(\_), rescale()) : range.slice();

};

scale.rangeRound = function(\_) {

return range = Array.from(\_), interpolate$1 = interpolateRound, rescale();

};

scale.clamp = function(\_) {

return arguments.length ? (clamp = \_ ? true : identity$1, rescale()) : clamp !== identity$1;

};

scale.interpolate = function(\_) {

return arguments.length ? (interpolate$1 = \_, rescale()) : interpolate$1;

};

scale.unknown = function(\_) {

return arguments.length ? (unknown = \_, scale) : unknown;

};

return function(t, u) {

transform = t, untransform = u;

return rescale();

};

}

function continuous() {

return transformer$1()(identity$1, identity$1);

}

function formatDecimal(x) {

return Math.abs(x = Math.round(x)) >= 1e21

? x.toLocaleString("en").replace(/,/g, "")

: x.toString(10);

}

function formatDecimalParts(x, p) {

if ((i = (x = p ? x.toExponential(p - 1) : x.toExponential()).indexOf("e")) < 0) return null; // NaN, ±Infinity

var i, coefficient = x.slice(0, i);

return [

coefficient.length > 1 ? coefficient[0] + coefficient.slice(2) : coefficient,

+x.slice(i + 1)

];

}

function exponent(x) {

return x = formatDecimalParts(Math.abs(x)), x ? x[1] : NaN;

}

function formatGroup(grouping, thousands) {

return function(value, width) {

var i = value.length,

t = [],

j = 0,

g = grouping[0],

length = 0;

while (i > 0 && g > 0) {

if (length + g + 1 > width) g = Math.max(1, width - length);

t.push(value.substring(i -= g, i + g));

if ((length += g + 1) > width) break;

g = grouping[j = (j + 1) % grouping.length];

}

return t.reverse().join(thousands);

};

}

function formatNumerals(numerals) {

return function(value) {

return value.replace(/[0-9]/g, function(i) {

return numerals[+i];

});

};

}

var re = /^(?:(.)?([<>=^]))?([+\-( ])?([$#])?(0)?(\d+)?(,)?(\.\d+)?(~)?([a-z%])?$/i;

function formatSpecifier(specifier) {

if (!(match = re.exec(specifier))) throw new Error("invalid format: " + specifier);

var match;

return new FormatSpecifier({

fill: match[1],

align: match[2],

sign: match[3],

symbol: match[4],

zero: match[5],

width: match[6],

comma: match[7],

precision: match[8] && match[8].slice(1),

trim: match[9],

type: match[10]

});

}

formatSpecifier.prototype = FormatSpecifier.prototype; // instanceof

function FormatSpecifier(specifier) {

this.fill = specifier.fill === undefined ? " " : specifier.fill + "";

this.align = specifier.align === undefined ? ">" : specifier.align + "";

this.sign = specifier.sign === undefined ? "-" : specifier.sign + "";

this.symbol = specifier.symbol === undefined ? "" : specifier.symbol + "";

this.zero = !!specifier.zero;

this.width = specifier.width === undefined ? undefined : +specifier.width;

this.comma = !!specifier.comma;

this.precision = specifier.precision === undefined ? undefined : +specifier.precision;

this.trim = !!specifier.trim;

this.type = specifier.type === undefined ? "" : specifier.type + "";

}

FormatSpecifier.prototype.toString = function() {

return this.fill

+ this.align

+ this.sign

+ this.symbol

+ (this.zero ? "0" : "")

+ (this.width === undefined ? "" : Math.max(1, this.width | 0))

+ (this.comma ? "," : "")

+ (this.precision === undefined ? "" : "." + Math.max(0, this.precision | 0))

+ (this.trim ? "~" : "")

+ this.type;

};

function formatTrim(s) {

out: for (var n = s.length, i = 1, i0 = -1, i1; i < n; ++i) {

switch (s[i]) {

case ".": i0 = i1 = i; break;

case "0": if (i0 === 0) i0 = i; i1 = i; break;

default: if (!+s[i]) break out; if (i0 > 0) i0 = 0; break;

}

}

return i0 > 0 ? s.slice(0, i0) + s.slice(i1 + 1) : s;

}

var prefixExponent;

function formatPrefixAuto(x, p) {

var d = formatDecimalParts(x, p);

if (!d) return x + "";

var coefficient = d[0],

exponent = d[1],

i = exponent - (prefixExponent = Math.max(-8, Math.min(8, Math.floor(exponent / 3))) \* 3) + 1,

n = coefficient.length;

return i === n ? coefficient

: i > n ? coefficient + new Array(i - n + 1).join("0")

: i > 0 ? coefficient.slice(0, i) + "." + coefficient.slice(i)

: "0." + new Array(1 - i).join("0") + formatDecimalParts(x, Math.max(0, p + i - 1))[0]; // less than 1y!

}

function formatRounded(x, p) {

var d = formatDecimalParts(x, p);

if (!d) return x + "";

var coefficient = d[0],

exponent = d[1];

return exponent < 0 ? "0." + new Array(-exponent).join("0") + coefficient

: coefficient.length > exponent + 1 ? coefficient.slice(0, exponent + 1) + "." + coefficient.slice(exponent + 1)

: coefficient + new Array(exponent - coefficient.length + 2).join("0");

}

var formatTypes = {

"%": (x, p) => (x \* 100).toFixed(p),

"b": (x) => Math.round(x).toString(2),

"c": (x) => x + "",

"d": formatDecimal,

"e": (x, p) => x.toExponential(p),

"f": (x, p) => x.toFixed(p),

"g": (x, p) => x.toPrecision(p),

"o": (x) => Math.round(x).toString(8),

"p": (x, p) => formatRounded(x \* 100, p),

"r": formatRounded,

"s": formatPrefixAuto,

"X": (x) => Math.round(x).toString(16).toUpperCase(),

"x": (x) => Math.round(x).toString(16)

};

function identity(x) {

return x;

}

var map = Array.prototype.map,

prefixes = ["y","z","a","f","p","n","µ","m","","k","M","G","T","P","E","Z","Y"];

function formatLocale(locale) {

var group = locale.grouping === undefined || locale.thousands === undefined ? identity : formatGroup(map.call(locale.grouping, Number), locale.thousands + ""),

currencyPrefix = locale.currency === undefined ? "" : locale.currency[0] + "",

currencySuffix = locale.currency === undefined ? "" : locale.currency[1] + "",

decimal = locale.decimal === undefined ? "." : locale.decimal + "",

numerals = locale.numerals === undefined ? identity : formatNumerals(map.call(locale.numerals, String)),

percent = locale.percent === undefined ? "%" : locale.percent + "",

minus = locale.minus === undefined ? "−" : locale.minus + "",

nan = locale.nan === undefined ? "NaN" : locale.nan + "";

function newFormat(specifier) {

specifier = formatSpecifier(specifier);

var fill = specifier.fill,

align = specifier.align,

sign = specifier.sign,

symbol = specifier.symbol,

zero = specifier.zero,

width = specifier.width,

comma = specifier.comma,

precision = specifier.precision,

trim = specifier.trim,

type = specifier.type;

if (type === "n") comma = true, type = "g";

else if (!formatTypes[type]) precision === undefined && (precision = 12), trim = true, type = "g";

if (zero || (fill === "0" && align === "=")) zero = true, fill = "0", align = "=";

var prefix = symbol === "$" ? currencyPrefix : symbol === "#" && /[boxX]/.test(type) ? "0" + type.toLowerCase() : "",

suffix = symbol === "$" ? currencySuffix : /[%p]/.test(type) ? percent : "";

var formatType = formatTypes[type],

maybeSuffix = /[defgprs%]/.test(type);

precision = precision === undefined ? 6

: /[gprs]/.test(type) ? Math.max(1, Math.min(21, precision))

: Math.max(0, Math.min(20, precision));

function format(value) {

var valuePrefix = prefix,

valueSuffix = suffix,

i, n, c;

if (type === "c") {

valueSuffix = formatType(value) + valueSuffix;

value = "";

} else {

value = +value;

var valueNegative = value < 0 || 1 / value < 0;

value = isNaN(value) ? nan : formatType(Math.abs(value), precision);

if (trim) value = formatTrim(value);

if (valueNegative && +value === 0 && sign !== "+") valueNegative = false;

valuePrefix = (valueNegative ? (sign === "(" ? sign : minus) : sign === "-" || sign === "(" ? "" : sign) + valuePrefix;

valueSuffix = (type === "s" ? prefixes[8 + prefixExponent / 3] : "") + valueSuffix + (valueNegative && sign === "(" ? ")" : "");

if (maybeSuffix) {

i = -1, n = value.length;

while (++i < n) {

if (c = value.charCodeAt(i), 48 > c || c > 57) {

valueSuffix = (c === 46 ? decimal + value.slice(i + 1) : value.slice(i)) + valueSuffix;

value = value.slice(0, i);

break;

}

}

}

}

if (comma && !zero) value = group(value, Infinity);

var length = valuePrefix.length + value.length + valueSuffix.length,

padding = length < width ? new Array(width - length + 1).join(fill) : "";

if (comma && zero) value = group(padding + value, padding.length ? width - valueSuffix.length : Infinity), padding = "";

switch (align) {

case "<": value = valuePrefix + value + valueSuffix + padding; break;

case "=": value = valuePrefix + padding + value + valueSuffix; break;

case "^": value = padding.slice(0, length = padding.length >> 1) + valuePrefix + value + valueSuffix + padding.slice(length); break;

default: value = padding + valuePrefix + value + valueSuffix; break;

}

return numerals(value);

}

format.toString = function() {

return specifier + "";

};

return format;

}

function formatPrefix(specifier, value) {

var f = newFormat((specifier = formatSpecifier(specifier), specifier.type = "f", specifier)),

e = Math.max(-8, Math.min(8, Math.floor(exponent(value) / 3))) \* 3,

k = Math.pow(10, -e),

prefix = prefixes[8 + e / 3];

return function(value) {

return f(k \* value) + prefix;

};

}

return {

format: newFormat,

formatPrefix: formatPrefix

};

}

var locale;

var format;

var formatPrefix;

defaultLocale({

thousands: ",",

grouping: [3],

currency: ["$", ""]

});

function defaultLocale(definition) {

locale = formatLocale(definition);

format = locale.format;

formatPrefix = locale.formatPrefix;

return locale;

}

function precisionFixed(step) {

return Math.max(0, -exponent(Math.abs(step)));

}

function precisionPrefix(step, value) {

return Math.max(0, Math.max(-8, Math.min(8, Math.floor(exponent(value) / 3))) \* 3 - exponent(Math.abs(step)));

}

function precisionRound(step, max) {

step = Math.abs(step), max = Math.abs(max) - step;

return Math.max(0, exponent(max) - exponent(step)) + 1;

}

function tickFormat(start, stop, count, specifier) {

var step = tickStep(start, stop, count),

precision;

specifier = formatSpecifier(specifier == null ? ",f" : specifier);

switch (specifier.type) {

case "s": {

var value = Math.max(Math.abs(start), Math.abs(stop));

if (specifier.precision == null && !isNaN(precision = precisionPrefix(step, value))) specifier.precision = precision;

return formatPrefix(specifier, value);

}

case "":

case "e":

case "g":

case "p":

case "r": {

if (specifier.precision == null && !isNaN(precision = precisionRound(step, Math.max(Math.abs(start), Math.abs(stop))))) specifier.precision = precision - (specifier.type === "e");

break;

}

case "f":

case "%": {

if (specifier.precision == null && !isNaN(precision = precisionFixed(step))) specifier.precision = precision - (specifier.type === "%") \* 2;

break;

}

}

return format(specifier);

}

function linearish(scale) {

var domain = scale.domain;

scale.ticks = function(count) {

var d = domain();

return ticks(d[0], d[d.length - 1], count == null ? 10 : count);

};

scale.tickFormat = function(count, specifier) {

var d = domain();

return tickFormat(d[0], d[d.length - 1], count == null ? 10 : count, specifier);

};

scale.nice = function(count) {

if (count == null) count = 10;

var d = domain();

var i0 = 0;

var i1 = d.length - 1;

var start = d[i0];

var stop = d[i1];

var prestep;

var step;

var maxIter = 10;

if (stop < start) {

step = start, start = stop, stop = step;

step = i0, i0 = i1, i1 = step;

}

while (maxIter-- > 0) {

step = tickIncrement(start, stop, count);

if (step === prestep) {

d[i0] = start;

d[i1] = stop;

return domain(d);

} else if (step > 0) {

start = Math.floor(start / step) \* step;

stop = Math.ceil(stop / step) \* step;

} else if (step < 0) {

start = Math.ceil(start \* step) / step;

stop = Math.floor(stop \* step) / step;

} else {

break;

}

prestep = step;

}

return scale;

};

return scale;

}

function linear() {

var scale = continuous();

scale.copy = function() {

return copy$1(scale, linear());

};

initRange.apply(scale, arguments);

return linearish(scale);

}

function transformer() {

var x0 = 0,

x1 = 1,

t0,

t1,

k10,

transform,

interpolator = identity$1,

clamp = false,

unknown;

function scale(x) {

return x == null || isNaN(x = +x) ? unknown : interpolator(k10 === 0 ? 0.5 : (x = (transform(x) - t0) \* k10, clamp ? Math.max(0, Math.min(1, x)) : x));

}

scale.domain = function(\_) {

return arguments.length ? ([x0, x1] = \_, t0 = transform(x0 = +x0), t1 = transform(x1 = +x1), k10 = t0 === t1 ? 0 : 1 / (t1 - t0), scale) : [x0, x1];

};

scale.clamp = function(\_) {

return arguments.length ? (clamp = !!\_, scale) : clamp;

};

scale.interpolator = function(\_) {

return arguments.length ? (interpolator = \_, scale) : interpolator;

};

function range(interpolate) {

return function(\_) {

var r0, r1;

return arguments.length ? ([r0, r1] = \_, interpolator = interpolate(r0, r1), scale) : [interpolator(0), interpolator(1)];

};

}

scale.range = range(interpolate);

scale.rangeRound = range(interpolateRound);

scale.unknown = function(\_) {

return arguments.length ? (unknown = \_, scale) : unknown;

};

return function(t) {

transform = t, t0 = t(x0), t1 = t(x1), k10 = t0 === t1 ? 0 : 1 / (t1 - t0);

return scale;

};

}

function copy(source, target) {

return target

.domain(source.domain())

.interpolator(source.interpolator())

.clamp(source.clamp())

.unknown(source.unknown());

}

function sequential() {

var scale = linearish(transformer()(identity$1));

scale.copy = function() {

return copy(scale, sequential());

};

return initInterpolator.apply(scale, arguments);

}

const COLOR\_BASE = "#cecece";

const rc = 0.2126;

const gc = 0.7152;

const bc = 0.0722;

const lowc = 1 / 12.92;

function adjustGamma(p) {

return Math.pow((p + 0.055) / 1.055, 2.4);

}

function relativeLuminance(o) {

const rsrgb = o.r / 255;

const gsrgb = o.g / 255;

const bsrgb = o.b / 255;

const r = rsrgb <= 0.03928 ? rsrgb \* lowc : adjustGamma(rsrgb);

const g = gsrgb <= 0.03928 ? gsrgb \* lowc : adjustGamma(gsrgb);

const b = bsrgb <= 0.03928 ? bsrgb \* lowc : adjustGamma(bsrgb);

return r \* rc + g \* gc + b \* bc;

}

const createRainbowColor = (root) => {

const colorParentMap = new Map();

colorParentMap.set(root, COLOR\_BASE);

if (root.children != null) {

const colorScale = sequential([0, root.children.length], (n) => hsl(360 \* n, 0.3, 0.85));

root.children.forEach((c, id) => {

colorParentMap.set(c, colorScale(id).toString());

});

}

const colorMap = new Map();

const lightScale = linear().domain([0, root.height]).range([0.9, 0.3]);

const getBackgroundColor = (node) => {

const parents = node.ancestors();

const colorStr = parents.length === 1

? colorParentMap.get(parents[0])

: colorParentMap.get(parents[parents.length - 2]);

const hslColor = hsl(colorStr);

hslColor.l = lightScale(node.depth);

return hslColor;

};

return (node) => {

if (!colorMap.has(node)) {

const backgroundColor = getBackgroundColor(node);

const l = relativeLuminance(backgroundColor.rgb());

const fontColor = l > 0.19 ? "#000" : "#fff";

colorMap.set(node, {

backgroundColor: backgroundColor.toString(),

fontColor,

});

}

return colorMap.get(node);

};

};

const StaticContext = J({});

const drawChart = (parentNode, data, width, height) => {

const availableSizeProperties = getAvailableSizeOptions(data.options);

console.time("layout create");

const layout = treemap()

.size([width, height])

.paddingOuter(PADDING)

.paddingTop(TOP\_PADDING)

.paddingInner(PADDING)

.round(true)

.tile(treemapResquarify);

console.timeEnd("layout create");

console.time("rawHierarchy create");

const rawHierarchy = hierarchy(data.tree);

console.timeEnd("rawHierarchy create");

const nodeSizesCache = new Map();

const nodeIdsCache = new Map();

const getModuleSize = (node, sizeKey) => { var \_a, \_b; return (\_b = (\_a = nodeSizesCache.get(node)) === null || \_a === void 0 ? void 0 : \_a[sizeKey]) !== null && \_b !== void 0 ? \_b : 0; };

console.time("rawHierarchy eachAfter cache");

rawHierarchy.eachAfter((node) => {

var \_a;

const nodeData = node.data;

nodeIdsCache.set(nodeData, {

nodeUid: generateUniqueId("node"),

clipUid: generateUniqueId("clip"),

});

const sizes = { renderedLength: 0, gzipLength: 0, brotliLength: 0 };

if (isModuleTree(nodeData)) {

for (const sizeKey of availableSizeProperties) {

sizes[sizeKey] = nodeData.children.reduce((acc, child) => getModuleSize(child, sizeKey) + acc, 0);

}

}

else {

for (const sizeKey of availableSizeProperties) {

sizes[sizeKey] = (\_a = data.nodeParts[nodeData.uid][sizeKey]) !== null && \_a !== void 0 ? \_a : 0;

}

}

nodeSizesCache.set(nodeData, sizes);

});

console.timeEnd("rawHierarchy eachAfter cache");

const getModuleIds = (node) => nodeIdsCache.get(node);

console.time("color");

const getModuleColor = createRainbowColor(rawHierarchy);

console.timeEnd("color");

D$1(u$1(StaticContext.Provider, { value: {

data,

availableSizeProperties,

width,

height,

getModuleSize,

getModuleIds,

getModuleColor,

rawHierarchy,

layout,

}, children: u$1(Main, {}) }), parentNode);

};

exports.StaticContext = StaticContext;

exports.default = drawChart;

Object.defineProperty(exports, '\_\_esModule', { value: true });

return exports;

})({});

/\*-->\*/

</script>

<script>

/\*<!--\*/

const data = {"version":2,"tree":{"name":"root","children":[{"name":"stream.js","children":[{"name":"src/stream.ts","uid":"7f425886-1"}]},{"name":"index.js","children":[{"name":"src","children":[{"uid":"7f425886-3","name":"errors.ts"},{"uid":"7f425886-5","name":"parse.ts"},{"uid":"7f425886-7","name":"index.ts"}]}]}],"isRoot":true},"nodeParts":{"7f425886-1":{"renderedLength":566,"gzipLength":275,"brotliLength":0,"metaUid":"7f425886-0"},"7f425886-3":{"renderedLength":226,"gzipLength":144,"brotliLength":0,"metaUid":"7f425886-2"},"7f425886-5":{"renderedLength":3359,"gzipLength":1213,"brotliLength":0,"metaUid":"7f425886-4"},"7f425886-7":{"renderedLength":0,"gzipLength":0,"brotliLength":0,"metaUid":"7f425886-6"}},"nodeMetas":{"7f425886-0":{"id":"/src/stream.ts","moduleParts":{"stream.js":"7f425886-1"},"imported":[{"uid":"7f425886-4"},{"uid":"7f425886-2"}],"importedBy":[],"isEntry":true},"7f425886-2":{"id":"/src/errors.ts","moduleParts":{"index.js":"7f425886-3"},"imported":[],"importedBy":[{"uid":"7f425886-0"},{"uid":"7f425886-4"},{"uid":"7f425886-6"}]},"7f425886-4":{"id":"/src/parse.ts","moduleParts":{"index.js":"7f425886-5"},"imported":[{"uid":"7f425886-2"}],"importedBy":[{"uid":"7f425886-0"},{"uid":"7f425886-6"}]},"7f425886-6":{"id":"/src/index.ts","moduleParts":{"index.js":"7f425886-7"},"imported":[{"uid":"7f425886-2"},{"uid":"7f425886-4"}],"importedBy":[],"isEntry":true}},"env":{"rollup":"4.40.2"},"options":{"gzip":true,"brotli":false,"sourcemap":false}};

const run = () => {

const width = window.innerWidth;

const height = window.innerHeight;

const chartNode = document.querySelector("main");

drawChart.default(chartNode, data, width, height);

};

window.addEventListener('resize', run);

document.addEventListener('DOMContentLoaded', run);

/\*-->\*/

</script>

</body>

</html>

/\* target file for webpack loader \*/

/\* target file for webpack loader \*/

@layer theme, base, components, utilities;

@layer theme {

@theme default {

--font-sans:

ui-sans-serif, system-ui, sans-serif, "Apple Color Emoji",

"Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";

--font-serif: ui-serif, Georgia, Cambria, "Times New Roman", Times, serif;

--font-mono:

ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas, "Liberation Mono",

"Courier New", monospace;

--color-red-50: oklch(97.1% 0.013 17.38);

--color-red-100: oklch(93.6% 0.032 17.717);

--color-red-200: oklch(88.5% 0.062 18.334);

--color-red-300: oklch(80.8% 0.114 19.571);

--color-red-400: oklch(70.4% 0.191 22.216);

--color-red-500: oklch(63.7% 0.237 25.331);

--color-red-600: oklch(57.7% 0.245 27.325);

--color-red-700: oklch(50.5% 0.213 27.518);

--color-red-800: oklch(44.4% 0.177 26.899);

--color-red-900: oklch(39.6% 0.141 25.723);

--color-red-950: oklch(25.8% 0.092 26.042);

--color-orange-50: oklch(98% 0.016 73.684);

--color-orange-100: oklch(95.4% 0.038 75.164);

--color-orange-200: oklch(90.1% 0.076 70.697);

--color-orange-300: oklch(83.7% 0.128 66.29);

--color-orange-400: oklch(75% 0.183 55.934);

--color-orange-500: oklch(70.5% 0.213 47.604);

--color-orange-600: oklch(64.6% 0.222 41.116);

--color-orange-700: oklch(55.3% 0.195 38.402);

--color-orange-800: oklch(47% 0.157 37.304);

--color-orange-900: oklch(40.8% 0.123 38.172);

--color-orange-950: oklch(26.6% 0.079 36.259);

--color-amber-50: oklch(98.7% 0.022 95.277);

--color-amber-100: oklch(96.2% 0.059 95.617);

--color-amber-200: oklch(92.4% 0.12 95.746);

--color-amber-300: oklch(87.9% 0.169 91.605);

--color-amber-400: oklch(82.8% 0.189 84.429);

--color-amber-500: oklch(76.9% 0.188 70.08);

--color-amber-600: oklch(66.6% 0.179 58.318);

--color-amber-700: oklch(55.5% 0.163 48.998);

--color-amber-800: oklch(47.3% 0.137 46.201);

--color-amber-900: oklch(41.4% 0.112 45.904);

--color-amber-950: oklch(27.9% 0.077 45.635);

--color-yellow-50: oklch(98.7% 0.026 102.212);

--color-yellow-100: oklch(97.3% 0.071 103.193);

--color-yellow-200: oklch(94.5% 0.129 101.54);

--color-yellow-300: oklch(90.5% 0.182 98.111);

--color-yellow-400: oklch(85.2% 0.199 91.936);

--color-yellow-500: oklch(79.5% 0.184 86.047);

--color-yellow-600: oklch(68.1% 0.162 75.834);

--color-yellow-700: oklch(55.4% 0.135 66.442);

--color-yellow-800: oklch(47.6% 0.114 61.907);

--color-yellow-900: oklch(42.1% 0.095 57.708);

--color-yellow-950: oklch(28.6% 0.066 53.813);

--color-lime-50: oklch(98.6% 0.031 120.757);

--color-lime-100: oklch(96.7% 0.067 122.328);

--color-lime-200: oklch(93.8% 0.127 124.321);

--color-lime-300: oklch(89.7% 0.196 126.665);

--color-lime-400: oklch(84.1% 0.238 128.85);

--color-lime-500: oklch(76.8% 0.233 130.85);

--color-lime-600: oklch(64.8% 0.2 131.684);

--color-lime-700: oklch(53.2% 0.157 131.589);

--color-lime-800: oklch(45.3% 0.124 130.933);

--color-lime-900: oklch(40.5% 0.101 131.063);

--color-lime-950: oklch(27.4% 0.072 132.109);

--color-green-50: oklch(98.2% 0.018 155.826);

--color-green-100: oklch(96.2% 0.044 156.743);

--color-green-200: oklch(92.5% 0.084 155.995);

--color-green-300: oklch(87.1% 0.15 154.449);

--color-green-400: oklch(79.2% 0.209 151.711);

--color-green-500: oklch(72.3% 0.219 149.579);

--color-green-600: oklch(62.7% 0.194 149.214);

--color-green-700: oklch(52.7% 0.154 150.069);

--color-green-800: oklch(44.8% 0.119 151.328);

--color-green-900: oklch(39.3% 0.095 152.535);

--color-green-950: oklch(26.6% 0.065 152.934);

--color-emerald-50: oklch(97.9% 0.021 166.113);

--color-emerald-100: oklch(95% 0.052 163.051);

--color-emerald-200: oklch(90.5% 0.093 164.15);

--color-emerald-300: oklch(84.5% 0.143 164.978);

--color-emerald-400: oklch(76.5% 0.177 163.223);

--color-emerald-500: oklch(69.6% 0.17 162.48);

--color-emerald-600: oklch(59.6% 0.145 163.225);

--color-emerald-700: oklch(50.8% 0.118 165.612);

--color-emerald-800: oklch(43.2% 0.095 166.913);

--color-emerald-900: oklch(37.8% 0.077 168.94);

--color-emerald-950: oklch(26.2% 0.051 172.552);

--color-teal-50: oklch(98.4% 0.014 180.72);

--color-teal-100: oklch(95.3% 0.051 180.801);

--color-teal-200: oklch(91% 0.096 180.426);

--color-teal-300: oklch(85.5% 0.138 181.071);

--color-teal-400: oklch(77.7% 0.152 181.912);

--color-teal-500: oklch(70.4% 0.14 182.503);

--color-teal-600: oklch(60% 0.118 184.704);

--color-teal-700: oklch(51.1% 0.096 186.391);

--color-teal-800: oklch(43.7% 0.078 188.216);

--color-teal-900: oklch(38.6% 0.063 188.416);

--color-teal-950: oklch(27.7% 0.046 192.524);

--color-cyan-50: oklch(98.4% 0.019 200.873);

--color-cyan-100: oklch(95.6% 0.045 203.388);

--color-cyan-200: oklch(91.7% 0.08 205.041);

--color-cyan-300: oklch(86.5% 0.127 207.078);

--color-cyan-400: oklch(78.9% 0.154 211.53);

--color-cyan-500: oklch(71.5% 0.143 215.221);

--color-cyan-600: oklch(60.9% 0.126 221.723);

--color-cyan-700: oklch(52% 0.105 223.128);

--color-cyan-800: oklch(45% 0.085 224.283);

--color-cyan-900: oklch(39.8% 0.07 227.392);

--color-cyan-950: oklch(30.2% 0.056 229.695);

--color-sky-50: oklch(97.7% 0.013 236.62);

--color-sky-100: oklch(95.1% 0.026 236.824);

--color-sky-200: oklch(90.1% 0.058 230.902);

--color-sky-300: oklch(82.8% 0.111 230.318);

--color-sky-400: oklch(74.6% 0.16 232.661);

--color-sky-500: oklch(68.5% 0.169 237.323);

--color-sky-600: oklch(58.8% 0.158 241.966);

--color-sky-700: oklch(50% 0.134 242.749);

--color-sky-800: oklch(44.3% 0.11 240.79);

--color-sky-900: oklch(39.1% 0.09 240.876);

--color-sky-950: oklch(29.3% 0.066 243.157);

--color-blue-50: oklch(97% 0.014 254.604);

--color-blue-100: oklch(93.2% 0.032 255.585);

--color-blue-200: oklch(88.2% 0.059 254.128);

--color-blue-300: oklch(80.9% 0.105 251.813);

--color-blue-400: oklch(70.7% 0.165 254.624);

--color-blue-500: oklch(62.3% 0.214 259.815);

--color-blue-600: oklch(54.6% 0.245 262.881);

--color-blue-700: oklch(48.8% 0.243 264.376);

--color-blue-800: oklch(42.4% 0.199 265.638);

--color-blue-900: oklch(37.9% 0.146 265.522);

--color-blue-950: oklch(28.2% 0.091 267.935);

--color-indigo-50: oklch(96.2% 0.018 272.314);

--color-indigo-100: oklch(93% 0.034 272.788);

--color-indigo-200: oklch(87% 0.065 274.039);

--color-indigo-300: oklch(78.5% 0.115 274.713);

--color-indigo-400: oklch(67.3% 0.182 276.935);

--color-indigo-500: oklch(58.5% 0.233 277.117);

--color-indigo-600: oklch(51.1% 0.262 276.966);

--color-indigo-700: oklch(45.7% 0.24 277.023);

--color-indigo-800: oklch(39.8% 0.195 277.366);

--color-indigo-900: oklch(35.9% 0.144 278.697);

--color-indigo-950: oklch(25.7% 0.09 281.288);

--color-violet-50: oklch(96.9% 0.016 293.756);

--color-violet-100: oklch(94.3% 0.029 294.588);

--color-violet-200: oklch(89.4% 0.057 293.283);

--color-violet-300: oklch(81.1% 0.111 293.571);

--color-violet-400: oklch(70.2% 0.183 293.541);

--color-violet-500: oklch(60.6% 0.25 292.717);

--color-violet-600: oklch(54.1% 0.281 293.009);

--color-violet-700: oklch(49.1% 0.27 292.581);

--color-violet-800: oklch(43.2% 0.232 292.759);

--color-violet-900: oklch(38% 0.189 293.745);

--color-violet-950: oklch(28.3% 0.141 291.089);

--color-purple-50: oklch(97.7% 0.014 308.299);

--color-purple-100: oklch(94.6% 0.033 307.174);

--color-purple-200: oklch(90.2% 0.063 306.703);

--color-purple-300: oklch(82.7% 0.119 306.383);

--color-purple-400: oklch(71.4% 0.203 305.504);

--color-purple-500: oklch(62.7% 0.265 303.9);

--color-purple-600: oklch(55.8% 0.288 302.321);

--color-purple-700: oklch(49.6% 0.265 301.924);

--color-purple-800: oklch(43.8% 0.218 303.724);

--color-purple-900: oklch(38.1% 0.176 304.987);

--color-purple-950: oklch(29.1% 0.149 302.717);

--color-fuchsia-50: oklch(97.7% 0.017 320.058);

--color-fuchsia-100: oklch(95.2% 0.037 318.852);

--color-fuchsia-200: oklch(90.3% 0.076 319.62);

--color-fuchsia-300: oklch(83.3% 0.145 321.434);

--color-fuchsia-400: oklch(74% 0.238 322.16);

--color-fuchsia-500: oklch(66.7% 0.295 322.15);

--color-fuchsia-600: oklch(59.1% 0.293 322.896);

--color-fuchsia-700: oklch(51.8% 0.253 323.949);

--color-fuchsia-800: oklch(45.2% 0.211 324.591);

--color-fuchsia-900: oklch(40.1% 0.17 325.612);

--color-fuchsia-950: oklch(29.3% 0.136 325.661);

--color-pink-50: oklch(97.1% 0.014 343.198);

--color-pink-100: oklch(94.8% 0.028 342.258);

--color-pink-200: oklch(89.9% 0.061 343.231);

--color-pink-300: oklch(82.3% 0.12 346.018);

--color-pink-400: oklch(71.8% 0.202 349.761);

--color-pink-500: oklch(65.6% 0.241 354.308);

--color-pink-600: oklch(59.2% 0.249 0.584);

--color-pink-700: oklch(52.5% 0.223 3.958);

--color-pink-800: oklch(45.9% 0.187 3.815);

--color-pink-900: oklch(40.8% 0.153 2.432);

--color-pink-950: oklch(28.4% 0.109 3.907);

--color-rose-50: oklch(96.9% 0.015 12.422);

--color-rose-100: oklch(94.1% 0.03 12.58);

--color-rose-200: oklch(89.2% 0.058 10.001);

--color-rose-300: oklch(81% 0.117 11.638);

--color-rose-400: oklch(71.2% 0.194 13.428);

--color-rose-500: oklch(64.5% 0.246 16.439);

--color-rose-600: oklch(58.6% 0.253 17.585);

--color-rose-700: oklch(51.4% 0.222 16.935);

--color-rose-800: oklch(45.5% 0.188 13.697);

--color-rose-900: oklch(41% 0.159 10.272);

--color-rose-950: oklch(27.1% 0.105 12.094);

--color-slate-50: oklch(98.4% 0.003 247.858);

--color-slate-100: oklch(96.8% 0.007 247.896);

--color-slate-200: oklch(92.9% 0.013 255.508);

--color-slate-300: oklch(86.9% 0.022 252.894);

--color-slate-400: oklch(70.4% 0.04 256.788);

--color-slate-500: oklch(55.4% 0.046 257.417);

--color-slate-600: oklch(44.6% 0.043 257.281);

--color-slate-700: oklch(37.2% 0.044 257.287);

--color-slate-800: oklch(27.9% 0.041 260.031);

--color-slate-900: oklch(20.8% 0.042 265.755);

--color-slate-950: oklch(12.9% 0.042 264.695);

--color-gray-50: oklch(98.5% 0.002 247.839);

--color-gray-100: oklch(96.7% 0.003 264.542);

--color-gray-200: oklch(92.8% 0.006 264.531);

--color-gray-300: oklch(87.2% 0.01 258.338);

--color-gray-400: oklch(70.7% 0.022 261.325);

--color-gray-500: oklch(55.1% 0.027 264.364);

--color-gray-600: oklch(44.6% 0.03 256.802);

--color-gray-700: oklch(37.3% 0.034 259.733);

--color-gray-800: oklch(27.8% 0.033 256.848);

--color-gray-900: oklch(21% 0.034 264.665);

--color-gray-950: oklch(13% 0.028 261.692);

--color-zinc-50: oklch(98.5% 0 0);

--color-zinc-100: oklch(96.7% 0.001 286.375);

--color-zinc-200: oklch(92% 0.004 286.32);

--color-zinc-300: oklch(87.1% 0.006 286.286);

--color-zinc-400: oklch(70.5% 0.015 286.067);

--color-zinc-500: oklch(55.2% 0.016 285.938);

--color-zinc-600: oklch(44.2% 0.017 285.786);

--color-zinc-700: oklch(37% 0.013 285.805);

--color-zinc-800: oklch(27.4% 0.006 286.033);

--color-zinc-900: oklch(21% 0.006 285.885);

--color-zinc-950: oklch(14.1% 0.005 285.823);

--color-neutral-50: oklch(98.5% 0 0);

--color-neutral-100: oklch(97% 0 0);

--color-neutral-200: oklch(92.2% 0 0);

--color-neutral-300: oklch(87% 0 0);

--color-neutral-400: oklch(70.8% 0 0);

--color-neutral-500: oklch(55.6% 0 0);

--color-neutral-600: oklch(43.9% 0 0);

--color-neutral-700: oklch(37.1% 0 0);

--color-neutral-800: oklch(26.9% 0 0);

--color-neutral-900: oklch(20.5% 0 0);

--color-neutral-950: oklch(14.5% 0 0);

--color-stone-50: oklch(98.5% 0.001 106.423);

--color-stone-100: oklch(97% 0.001 106.424);

--color-stone-200: oklch(92.3% 0.003 48.717);

--color-stone-300: oklch(86.9% 0.005 56.366);

--color-stone-400: oklch(70.9% 0.01 56.259);

--color-stone-500: oklch(55.3% 0.013 58.071);

--color-stone-600: oklch(44.4% 0.011 73.639);

--color-stone-700: oklch(37.4% 0.01 67.558);

--color-stone-800: oklch(26.8% 0.007 34.298);

--color-stone-900: oklch(21.6% 0.006 56.043);

--color-stone-950: oklch(14.7% 0.004 49.25);

--color-black: #000;

--color-white: #fff;

--spacing: 0.25rem;

--breakpoint-sm: 40rem;

--breakpoint-md: 48rem;

--breakpoint-lg: 64rem;

--breakpoint-xl: 80rem;

--breakpoint-2xl: 96rem;

--container-3xs: 16rem;

--container-2xs: 18rem;

--container-xs: 20rem;

--container-sm: 24rem;

--container-md: 28rem;

--container-lg: 32rem;

--container-xl: 36rem;

--container-2xl: 42rem;

--container-3xl: 48rem;

--container-4xl: 56rem;

--container-5xl: 64rem;

--container-6xl: 72rem;

--container-7xl: 80rem;

--text-xs: 0.75rem;

--text-xs--line-height: calc(1 / 0.75);

--text-sm: 0.875rem;

--text-sm--line-height: calc(1.25 / 0.875);

--text-base: 1rem;

--text-base--line-height: calc(1.5 / 1);

--text-lg: 1.125rem;

--text-lg--line-height: calc(1.75 / 1.125);

--text-xl: 1.25rem;

--text-xl--line-height: calc(1.75 / 1.25);

--text-2xl: 1.5rem;

--text-2xl--line-height: calc(2 / 1.5);

--text-3xl: 1.875rem;

--text-3xl--line-height: calc(2.25 / 1.875);

--text-4xl: 2.25rem;

--text-4xl--line-height: calc(2.5 / 2.25);

--text-5xl: 3rem;

--text-5xl--line-height: 1;

--text-6xl: 3.75rem;

--text-6xl--line-height: 1;

--text-7xl: 4.5rem;

--text-7xl--line-height: 1;

--text-8xl: 6rem;

--text-8xl--line-height: 1;

--text-9xl: 8rem;

--text-9xl--line-height: 1;

--font-weight-thin: 100;

--font-weight-extralight: 200;

--font-weight-light: 300;

--font-weight-normal: 400;

--font-weight-medium: 500;

--font-weight-semibold: 600;

--font-weight-bold: 700;

--font-weight-extrabold: 800;

--font-weight-black: 900;

--tracking-tighter: -0.05em;

--tracking-tight: -0.025em;

--tracking-normal: 0em;

--tracking-wide: 0.025em;

--tracking-wider: 0.05em;

--tracking-widest: 0.1em;

--leading-tight: 1.25;

--leading-snug: 1.375;

--leading-normal: 1.5;

--leading-relaxed: 1.625;

--leading-loose: 2;

--radius-xs: 0.125rem;

--radius-sm: 0.25rem;

--radius-md: 0.375rem;

--radius-lg: 0.5rem;

--radius-xl: 0.75rem;

--radius-2xl: 1rem;

--radius-3xl: 1.5rem;

--radius-4xl: 2rem;

--shadow-2xs: 0 1px rgb(0 0 0 / 0.05);

--shadow-xs: 0 1px 2px 0 rgb(0 0 0 / 0.05);

--shadow-sm: 0 1px 3px 0 rgb(0 0 0 / 0.1), 0 1px 2px -1px rgb(0 0 0 / 0.1);

--shadow-md:

0 4px 6px -1px rgb(0 0 0 / 0.1), 0 2px 4px -2px rgb(0 0 0 / 0.1);

--shadow-lg:

0 10px 15px -3px rgb(0 0 0 / 0.1), 0 4px 6px -4px rgb(0 0 0 / 0.1);

--shadow-xl:

0 20px 25px -5px rgb(0 0 0 / 0.1), 0 8px 10px -6px rgb(0 0 0 / 0.1);

--shadow-2xl: 0 25px 50px -12px rgb(0 0 0 / 0.25);

--inset-shadow-2xs: inset 0 1px rgb(0 0 0 / 0.05);

--inset-shadow-xs: inset 0 1px 1px rgb(0 0 0 / 0.05);

--inset-shadow-sm: inset 0 2px 4px rgb(0 0 0 / 0.05);

--drop-shadow-xs: 0 1px 1px rgb(0 0 0 / 0.05);

--drop-shadow-sm: 0 1px 2px rgb(0 0 0 / 0.15);

--drop-shadow-md: 0 3px 3px rgb(0 0 0 / 0.12);

--drop-shadow-lg: 0 4px 4px rgb(0 0 0 / 0.15);

--drop-shadow-xl: 0 9px 7px rgb(0 0 0 / 0.1);

--drop-shadow-2xl: 0 25px 25px rgb(0 0 0 / 0.15);

--text-shadow-2xs: 0px 1px 0px rgb(0 0 0 / 0.15);

--text-shadow-xs: 0px 1px 1px rgb(0 0 0 / 0.2);

--text-shadow-sm:

0px 1px 0px rgb(0 0 0 / 0.075), 0px 1px 1px rgb(0 0 0 / 0.075),

0px 2px 2px rgb(0 0 0 / 0.075);

--text-shadow-md:

0px 1px 1px rgb(0 0 0 / 0.1), 0px 1px 2px rgb(0 0 0 / 0.1),

0px 2px 4px rgb(0 0 0 / 0.1);

--text-shadow-lg:

0px 1px 2px rgb(0 0 0 / 0.1), 0px 3px 2px rgb(0 0 0 / 0.1),

0px 4px 8px rgb(0 0 0 / 0.1);

--ease-in: cubic-bezier(0.4, 0, 1, 1);

--ease-out: cubic-bezier(0, 0, 0.2, 1);

--ease-in-out: cubic-bezier(0.4, 0, 0.2, 1);

--animate-spin: spin 1s linear infinite;

--animate-ping: ping 1s cubic-bezier(0, 0, 0.2, 1) infinite;

--animate-pulse: pulse 2s cubic-bezier(0.4, 0, 0.6, 1) infinite;

--animate-bounce: bounce 1s infinite;

@keyframes spin {

to {

transform: rotate(360deg);

}

}

@keyframes ping {

75%,

100% {

transform: scale(2);

opacity: 0;

}

}

@keyframes pulse {

50% {

opacity: 0.5;

}

}

@keyframes bounce {

0%,

100% {

transform: translateY(-25%);

animation-timing-function: cubic-bezier(0.8, 0, 1, 1);

}

50% {

transform: none;

animation-timing-function: cubic-bezier(0, 0, 0.2, 1);

}

}

--blur-xs: 4px;

--blur-sm: 8px;

--blur-md: 12px;

--blur-lg: 16px;

--blur-xl: 24px;

--blur-2xl: 40px;

--blur-3xl: 64px;

--perspective-dramatic: 100px;

--perspective-near: 300px;

--perspective-normal: 500px;

--perspective-midrange: 800px;

--perspective-distant: 1200px;

--aspect-video: 16 / 9;

--default-transition-duration: 150ms;

--default-transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1);

--default-font-family: --theme(--font-sans, initial);

--default-font-feature-settings: --theme(

--font-sans--font-feature-settings,

initial

);

--default-font-variation-settings: --theme(

--font-sans--font-variation-settings,

initial

);

--default-mono-font-family: --theme(--font-mono, initial);

--default-mono-font-feature-settings: --theme(

--font-mono--font-feature-settings,

initial

);

--default-mono-font-variation-settings: --theme(

--font-mono--font-variation-settings,

initial

);

}

/\* Deprecated \*/

@theme default inline reference {

--blur: 8px;

--shadow: 0 1px 3px 0 rgb(0 0 0 / 0.1), 0 1px 2px -1px rgb(0 0 0 / 0.1);

--shadow-inner: inset 0 2px 4px 0 rgb(0 0 0 / 0.05);

--drop-shadow: 0 1px 2px rgb(0 0 0 / 0.1), 0 1px 1px rgb(0 0 0 / 0.06);

--radius: 0.25rem;

--max-width-prose: 65ch;

}

}

@layer base {

/\*

1. Prevent padding and border from affecting element width. (https://github.com/mozdevs/cssremedy/issues/4)

2. Remove default margins and padding

3. Reset all borders.

\*/

\*,

::after,

::before,

::backdrop,

::file-selector-button {

box-sizing: border-box; /\* 1 \*/

margin: 0; /\* 2 \*/

padding: 0; /\* 2 \*/

border: 0 solid; /\* 3 \*/

}

/\*

1. Use a consistent sensible line-height in all browsers.

2. Prevent adjustments of font size after orientation changes in iOS.

3. Use a more readable tab size.

4. Use the user's configured `sans` font-family by default.

5. Use the user's configured `sans` font-feature-settings by default.

6. Use the user's configured `sans` font-variation-settings by default.

7. Disable tap highlights on iOS.

\*/

html,

:host {

line-height: 1.5; /\* 1 \*/

-webkit-text-size-adjust: 100%; /\* 2 \*/

tab-size: 4; /\* 3 \*/

font-family: --theme(

--default-font-family,

ui-sans-serif,

system-ui,

sans-serif,

"Apple Color Emoji",

"Segoe UI Emoji",

"Segoe UI Symbol",

"Noto Color Emoji"

); /\* 4 \*/

font-feature-settings: --theme(

--default-font-feature-settings,

normal

); /\* 5 \*/

font-variation-settings: --theme(

--default-font-variation-settings,

normal

); /\* 6 \*/

-webkit-tap-highlight-color: transparent; /\* 7 \*/

}

/\*

1. Add the correct height in Firefox.

2. Correct the inheritance of border color in Firefox. (https://bugzilla.mozilla.org/show\_bug.cgi?id=190655)

3. Reset the default border style to a 1px solid border.

\*/

hr {

height: 0; /\* 1 \*/

color: inherit; /\* 2 \*/

border-top-width: 1px; /\* 3 \*/

}

/\*

Add the correct text decoration in Chrome, Edge, and Safari.

\*/

abbr:where([title]) {

-webkit-text-decoration: underline dotted;

text-decoration: underline dotted;

}

/\*

Remove the default font size and weight for headings.

\*/

h1,

h2,

h3,

h4,

h5,

h6 {

font-size: inherit;

font-weight: inherit;

}

/\*

Reset links to optimize for opt-in styling instead of opt-out.

\*/

a {

color: inherit;

-webkit-text-decoration: inherit;

text-decoration: inherit;

}

/\*

Add the correct font weight in Edge and Safari.

\*/

b,

strong {

font-weight: bolder;

}

/\*

1. Use the user's configured `mono` font-family by default.

2. Use the user's configured `mono` font-feature-settings by default.

3. Use the user's configured `mono` font-variation-settings by default.

4. Correct the odd `em` font sizing in all browsers.

\*/

code,

kbd,

samp,

pre {

font-family: --theme(

--default-mono-font-family,

ui-monospace,

SFMono-Regular,

Menlo,

Monaco,

Consolas,

"Liberation Mono",

"Courier New",

monospace

); /\* 1 \*/

font-feature-settings: --theme(

--default-mono-font-feature-settings,

normal

); /\* 2 \*/

font-variation-settings: --theme(

--default-mono-font-variation-settings,

normal

); /\* 3 \*/

font-size: 1em; /\* 4 \*/

}

/\*

Add the correct font size in all browsers.

\*/

small {

font-size: 80%;

}

/\*

Prevent `sub` and `sup` elements from affecting the line height in all browsers.

\*/

sub,

sup {

font-size: 75%;

line-height: 0;

position: relative;

vertical-align: baseline;

}

sub {

bottom: -0.25em;

}

sup {

top: -0.5em;

}

/\*

1. Remove text indentation from table contents in Chrome and Safari. (https://bugs.chromium.org/p/chromium/issues/detail?id=999088, https://bugs.webkit.org/show\_bug.cgi?id=201297)

2. Correct table border color inheritance in all Chrome and Safari. (https://bugs.chromium.org/p/chromium/issues/detail?id=935729, https://bugs.webkit.org/show\_bug.cgi?id=195016)

3. Remove gaps between table borders by default.

\*/

table {

text-indent: 0; /\* 1 \*/

border-color: inherit; /\* 2 \*/

border-collapse: collapse; /\* 3 \*/

}

/\*

Use the modern Firefox focus style for all focusable elements.

\*/

:-moz-focusring {

outline: auto;

}

/\*

Add the correct vertical alignment in Chrome and Firefox.

\*/

progress {

vertical-align: baseline;

}

/\*

Add the correct display in Chrome and Safari.

\*/

summary {

display: list-item;

}

/\*

Make lists unstyled by default.

\*/

ol,

ul,

menu {

list-style: none;

}

/\*

1. Make replaced elements `display: block` by default. (https://github.com/mozdevs/cssremedy/issues/14)

2. Add `vertical-align: middle` to align replaced elements more sensibly by default. (https://github.com/jensimmons/cssremedy/issues/14#issuecomment-634934210)

This can trigger a poorly considered lint error in some tools but is included by design.

\*/

img,

svg,

video,

canvas,

audio,

iframe,

embed,

object {

display: block; /\* 1 \*/

vertical-align: middle; /\* 2 \*/

}

/\*

Constrain images and videos to the parent width and preserve their intrinsic aspect ratio. (https://github.com/mozdevs/cssremedy/issues/14)

\*/

img,

video {

max-width: 100%;

height: auto;

}

/\*

1. Inherit font styles in all browsers.

2. Remove border radius in all browsers.

3. Remove background color in all browsers.

4. Ensure consistent opacity for disabled states in all browsers.

\*/

button,

input,

select,

optgroup,

textarea,

::file-selector-button {

font: inherit; /\* 1 \*/

font-feature-settings: inherit; /\* 1 \*/

font-variation-settings: inherit; /\* 1 \*/

letter-spacing: inherit; /\* 1 \*/

color: inherit; /\* 1 \*/

border-radius: 0; /\* 2 \*/

background-color: transparent; /\* 3 \*/

opacity: 1; /\* 4 \*/

}

/\*

Restore default font weight.

\*/

:where(select:is([multiple], [size])) optgroup {

font-weight: bolder;

}

/\*

Restore indentation.

\*/

:where(select:is([multiple], [size])) optgroup option {

padding-inline-start: 20px;

}

/\*

Restore space after button.

\*/

::file-selector-button {

margin-inline-end: 4px;

}

/\*

Reset the default placeholder opacity in Firefox. (https://github.com/tailwindlabs/tailwindcss/issues/3300)

\*/

::placeholder {

opacity: 1;

}

/\*

Set the default placeholder color to a semi-transparent version of the current text color in browsers that do not

crash when using `color-mix(…)` with `currentcolor`. (https://github.com/tailwindlabs/tailwindcss/issues/17194)

\*/

@supports (not (-webkit-appearance: -apple-pay-button)) /\* Not Safari \*/ or

(contain-intrinsic-size: 1px) /\* Safari 17+ \*/ {

::placeholder {

color: color-mix(in oklab, currentcolor 50%, transparent);

}

}

/\*

Prevent resizing textareas horizontally by default.

\*/

textarea {

resize: vertical;

}

/\*

Remove the inner padding in Chrome and Safari on macOS.

\*/

::-webkit-search-decoration {

-webkit-appearance: none;

}

/\*

1. Ensure date/time inputs have the same height when empty in iOS Safari.

2. Ensure text alignment can be changed on date/time inputs in iOS Safari.

\*/

::-webkit-date-and-time-value {

min-height: 1lh; /\* 1 \*/

text-align: inherit; /\* 2 \*/

}

/\*

Prevent height from changing on date/time inputs in macOS Safari when the input is set to `display: block`.

\*/

::-webkit-datetime-edit {

display: inline-flex;

}

/\*

Remove excess padding from pseudo-elements in date/time inputs to ensure consistent height across browsers.

\*/

::-webkit-datetime-edit-fields-wrapper {

padding: 0;

}

::-webkit-datetime-edit,

::-webkit-datetime-edit-year-field,

::-webkit-datetime-edit-month-field,

::-webkit-datetime-edit-day-field,

::-webkit-datetime-edit-hour-field,

::-webkit-datetime-edit-minute-field,

::-webkit-datetime-edit-second-field,

::-webkit-datetime-edit-millisecond-field,

::-webkit-datetime-edit-meridiem-field {

padding-block: 0;

}

/\*

Remove the additional `:invalid` styles in Firefox. (https://github.com/mozilla/gecko-dev/blob/2f9eacd9d3d995c937b4251a5557d95d494c9be1/layout/style/res/forms.css#L728-L737)

\*/

:-moz-ui-invalid {

box-shadow: none;

}

/\*

Correct the inability to style the border radius in iOS Safari.

\*/

button,

input:where([type="button"], [type="reset"], [type="submit"]),

::file-selector-button {

appearance: button;

}

/\*

Correct the cursor style of increment and decrement buttons in Safari.

\*/

::-webkit-inner-spin-button,

::-webkit-outer-spin-button {

height: auto;

}

/\*

Make elements with the HTML hidden attribute stay hidden by default.

\*/

[hidden]:where(:not([hidden="until-found"])) {

display: none !important;

}

}

@layer utilities {

@tailwind utilities;

}

/\*

1. Prevent padding and border from affecting element width. (https://github.com/mozdevs/cssremedy/issues/4)

2. Remove default margins and padding

3. Reset all borders.

\*/

\*,

::after,

::before,

::backdrop,

::file-selector-button {

box-sizing: border-box; /\* 1 \*/

margin: 0; /\* 2 \*/

padding: 0; /\* 2 \*/

border: 0 solid; /\* 3 \*/

}

/\*

1. Use a consistent sensible line-height in all browsers.

2. Prevent adjustments of font size after orientation changes in iOS.

3. Use a more readable tab size.

4. Use the user's configured `sans` font-family by default.

5. Use the user's configured `sans` font-feature-settings by default.

6. Use the user's configured `sans` font-variation-settings by default.

7. Disable tap highlights on iOS.

\*/

html,

:host {

line-height: 1.5; /\* 1 \*/

-webkit-text-size-adjust: 100%; /\* 2 \*/

tab-size: 4; /\* 3 \*/

font-family: --theme(

--default-font-family,

ui-sans-serif,

system-ui,

sans-serif,

'Apple Color Emoji',

'Segoe UI Emoji',

'Segoe UI Symbol',

'Noto Color Emoji'

); /\* 4 \*/

font-feature-settings: --theme(--default-font-feature-settings, normal); /\* 5 \*/

font-variation-settings: --theme(--default-font-variation-settings, normal); /\* 6 \*/

-webkit-tap-highlight-color: transparent; /\* 7 \*/

}

/\*

1. Add the correct height in Firefox.

2. Correct the inheritance of border color in Firefox. (https://bugzilla.mozilla.org/show\_bug.cgi?id=190655)

3. Reset the default border style to a 1px solid border.

\*/

hr {

height: 0; /\* 1 \*/

color: inherit; /\* 2 \*/

border-top-width: 1px; /\* 3 \*/

}

/\*

Add the correct text decoration in Chrome, Edge, and Safari.

\*/

abbr:where([title]) {

-webkit-text-decoration: underline dotted;

text-decoration: underline dotted;

}

/\*

Remove the default font size and weight for headings.

\*/

h1,

h2,

h3,

h4,

h5,

h6 {

font-size: inherit;

font-weight: inherit;

}

/\*

Reset links to optimize for opt-in styling instead of opt-out.

\*/

a {

color: inherit;

-webkit-text-decoration: inherit;

text-decoration: inherit;

}

/\*

Add the correct font weight in Edge and Safari.

\*/

b,

strong {

font-weight: bolder;

}

/\*

1. Use the user's configured `mono` font-family by default.

2. Use the user's configured `mono` font-feature-settings by default.

3. Use the user's configured `mono` font-variation-settings by default.

4. Correct the odd `em` font sizing in all browsers.

\*/

code,

kbd,

samp,

pre {

font-family: --theme(

--default-mono-font-family,

ui-monospace,

SFMono-Regular,

Menlo,

Monaco,

Consolas,

'Liberation Mono',

'Courier New',

monospace

); /\* 1 \*/

font-feature-settings: --theme(--default-mono-font-feature-settings, normal); /\* 2 \*/

font-variation-settings: --theme(--default-mono-font-variation-settings, normal); /\* 3 \*/

font-size: 1em; /\* 4 \*/

}

/\*

Add the correct font size in all browsers.

\*/

small {

font-size: 80%;

}

/\*

Prevent `sub` and `sup` elements from affecting the line height in all browsers.

\*/

sub,

sup {

font-size: 75%;

line-height: 0;

position: relative;

vertical-align: baseline;

}

sub {

bottom: -0.25em;

}

sup {

top: -0.5em;

}

/\*

1. Remove text indentation from table contents in Chrome and Safari. (https://bugs.chromium.org/p/chromium/issues/detail?id=999088, https://bugs.webkit.org/show\_bug.cgi?id=201297)

2. Correct table border color inheritance in all Chrome and Safari. (https://bugs.chromium.org/p/chromium/issues/detail?id=935729, https://bugs.webkit.org/show\_bug.cgi?id=195016)

3. Remove gaps between table borders by default.

\*/

table {

text-indent: 0; /\* 1 \*/

border-color: inherit; /\* 2 \*/

border-collapse: collapse; /\* 3 \*/

}

/\*

Use the modern Firefox focus style for all focusable elements.

\*/

:-moz-focusring {

outline: auto;

}

/\*

Add the correct vertical alignment in Chrome and Firefox.

\*/

progress {

vertical-align: baseline;

}

/\*

Add the correct display in Chrome and Safari.

\*/

summary {

display: list-item;

}

/\*

Make lists unstyled by default.

\*/

ol,

ul,

menu {

list-style: none;

}

/\*

1. Make replaced elements `display: block` by default. (https://github.com/mozdevs/cssremedy/issues/14)

2. Add `vertical-align: middle` to align replaced elements more sensibly by default. (https://github.com/jensimmons/cssremedy/issues/14#issuecomment-634934210)

This can trigger a poorly considered lint error in some tools but is included by design.

\*/

img,

svg,

video,

canvas,

audio,

iframe,

embed,

object {

display: block; /\* 1 \*/

vertical-align: middle; /\* 2 \*/

}

/\*

Constrain images and videos to the parent width and preserve their intrinsic aspect ratio. (https://github.com/mozdevs/cssremedy/issues/14)

\*/

img,

video {

max-width: 100%;

height: auto;

}

/\*

1. Inherit font styles in all browsers.

2. Remove border radius in all browsers.

3. Remove background color in all browsers.

4. Ensure consistent opacity for disabled states in all browsers.

\*/

button,

input,

select,

optgroup,

textarea,

::file-selector-button {

font: inherit; /\* 1 \*/

font-feature-settings: inherit; /\* 1 \*/

font-variation-settings: inherit; /\* 1 \*/

letter-spacing: inherit; /\* 1 \*/

color: inherit; /\* 1 \*/

border-radius: 0; /\* 2 \*/

background-color: transparent; /\* 3 \*/

opacity: 1; /\* 4 \*/

}

/\*

Restore default font weight.

\*/

:where(select:is([multiple], [size])) optgroup {

font-weight: bolder;

}

/\*

Restore indentation.

\*/

:where(select:is([multiple], [size])) optgroup option {

padding-inline-start: 20px;

}

/\*

Restore space after button.

\*/

::file-selector-button {

margin-inline-end: 4px;

}

/\*

Reset the default placeholder opacity in Firefox. (https://github.com/tailwindlabs/tailwindcss/issues/3300)

\*/

::placeholder {

opacity: 1;

}

/\*

Set the default placeholder color to a semi-transparent version of the current text color in browsers that do not

crash when using `color-mix(…)` with `currentcolor`. (https://github.com/tailwindlabs/tailwindcss/issues/17194)

\*/

@supports (not (-webkit-appearance: -apple-pay-button)) /\* Not Safari \*/ or

(contain-intrinsic-size: 1px) /\* Safari 17+ \*/ {

::placeholder {

color: color-mix(in oklab, currentcolor 50%, transparent);

}

}

/\*

Prevent resizing textareas horizontally by default.

\*/

textarea {

resize: vertical;

}

/\*

Remove the inner padding in Chrome and Safari on macOS.

\*/

::-webkit-search-decoration {

-webkit-appearance: none;

}

/\*

1. Ensure date/time inputs have the same height when empty in iOS Safari.

2. Ensure text alignment can be changed on date/time inputs in iOS Safari.

\*/

::-webkit-date-and-time-value {

min-height: 1lh; /\* 1 \*/

text-align: inherit; /\* 2 \*/

}

/\*

Prevent height from changing on date/time inputs in macOS Safari when the input is set to `display: block`.

\*/

::-webkit-datetime-edit {

display: inline-flex;

}

/\*

Remove excess padding from pseudo-elements in date/time inputs to ensure consistent height across browsers.

\*/

::-webkit-datetime-edit-fields-wrapper {

padding: 0;

}

::-webkit-datetime-edit,

::-webkit-datetime-edit-year-field,

::-webkit-datetime-edit-month-field,

::-webkit-datetime-edit-day-field,

::-webkit-datetime-edit-hour-field,

::-webkit-datetime-edit-minute-field,

::-webkit-datetime-edit-second-field,

::-webkit-datetime-edit-millisecond-field,

::-webkit-datetime-edit-meridiem-field {

padding-block: 0;

}

/\*

Remove the additional `:invalid` styles in Firefox. (https://github.com/mozilla/gecko-dev/blob/2f9eacd9d3d995c937b4251a5557d95d494c9be1/layout/style/res/forms.css#L728-L737)

\*/

:-moz-ui-invalid {

box-shadow: none;

}

/\*

Correct the inability to style the border radius in iOS Safari.

\*/

button,

input:where([type='button'], [type='reset'], [type='submit']),

::file-selector-button {

appearance: button;

}

/\*

Correct the cursor style of increment and decrement buttons in Safari.

\*/

::-webkit-inner-spin-button,

::-webkit-outer-spin-button {

height: auto;

}

/\*

Make elements with the HTML hidden attribute stay hidden by default.

\*/

[hidden]:where(:not([hidden='until-found'])) {

display: none !important;

}

@theme default {

--font-sans:

ui-sans-serif, system-ui, sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol',

'Noto Color Emoji';

--font-serif: ui-serif, Georgia, Cambria, 'Times New Roman', Times, serif;

--font-mono:

ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas, 'Liberation Mono', 'Courier New',

monospace;

--color-red-50: oklch(97.1% 0.013 17.38);

--color-red-100: oklch(93.6% 0.032 17.717);

--color-red-200: oklch(88.5% 0.062 18.334);

--color-red-300: oklch(80.8% 0.114 19.571);

--color-red-400: oklch(70.4% 0.191 22.216);

--color-red-500: oklch(63.7% 0.237 25.331);

--color-red-600: oklch(57.7% 0.245 27.325);

--color-red-700: oklch(50.5% 0.213 27.518);

--color-red-800: oklch(44.4% 0.177 26.899);

--color-red-900: oklch(39.6% 0.141 25.723);

--color-red-950: oklch(25.8% 0.092 26.042);

--color-orange-50: oklch(98% 0.016 73.684);

--color-orange-100: oklch(95.4% 0.038 75.164);

--color-orange-200: oklch(90.1% 0.076 70.697);

--color-orange-300: oklch(83.7% 0.128 66.29);

--color-orange-400: oklch(75% 0.183 55.934);

--color-orange-500: oklch(70.5% 0.213 47.604);

--color-orange-600: oklch(64.6% 0.222 41.116);

--color-orange-700: oklch(55.3% 0.195 38.402);

--color-orange-800: oklch(47% 0.157 37.304);

--color-orange-900: oklch(40.8% 0.123 38.172);

--color-orange-950: oklch(26.6% 0.079 36.259);

--color-amber-50: oklch(98.7% 0.022 95.277);

--color-amber-100: oklch(96.2% 0.059 95.617);

--color-amber-200: oklch(92.4% 0.12 95.746);

--color-amber-300: oklch(87.9% 0.169 91.605);

--color-amber-400: oklch(82.8% 0.189 84.429);

--color-amber-500: oklch(76.9% 0.188 70.08);

--color-amber-600: oklch(66.6% 0.179 58.318);

--color-amber-700: oklch(55.5% 0.163 48.998);

--color-amber-800: oklch(47.3% 0.137 46.201);

--color-amber-900: oklch(41.4% 0.112 45.904);

--color-amber-950: oklch(27.9% 0.077 45.635);

--color-yellow-50: oklch(98.7% 0.026 102.212);

--color-yellow-100: oklch(97.3% 0.071 103.193);

--color-yellow-200: oklch(94.5% 0.129 101.54);

--color-yellow-300: oklch(90.5% 0.182 98.111);

--color-yellow-400: oklch(85.2% 0.199 91.936);

--color-yellow-500: oklch(79.5% 0.184 86.047);

--color-yellow-600: oklch(68.1% 0.162 75.834);

--color-yellow-700: oklch(55.4% 0.135 66.442);

--color-yellow-800: oklch(47.6% 0.114 61.907);

--color-yellow-900: oklch(42.1% 0.095 57.708);

--color-yellow-950: oklch(28.6% 0.066 53.813);

--color-lime-50: oklch(98.6% 0.031 120.757);

--color-lime-100: oklch(96.7% 0.067 122.328);

--color-lime-200: oklch(93.8% 0.127 124.321);

--color-lime-300: oklch(89.7% 0.196 126.665);

--color-lime-400: oklch(84.1% 0.238 128.85);

--color-lime-500: oklch(76.8% 0.233 130.85);

--color-lime-600: oklch(64.8% 0.2 131.684);

--color-lime-700: oklch(53.2% 0.157 131.589);

--color-lime-800: oklch(45.3% 0.124 130.933);

--color-lime-900: oklch(40.5% 0.101 131.063);

--color-lime-950: oklch(27.4% 0.072 132.109);

--color-green-50: oklch(98.2% 0.018 155.826);

--color-green-100: oklch(96.2% 0.044 156.743);

--color-green-200: oklch(92.5% 0.084 155.995);

--color-green-300: oklch(87.1% 0.15 154.449);

--color-green-400: oklch(79.2% 0.209 151.711);

--color-green-500: oklch(72.3% 0.219 149.579);

--color-green-600: oklch(62.7% 0.194 149.214);

--color-green-700: oklch(52.7% 0.154 150.069);

--color-green-800: oklch(44.8% 0.119 151.328);

--color-green-900: oklch(39.3% 0.095 152.535);

--color-green-950: oklch(26.6% 0.065 152.934);

--color-emerald-50: oklch(97.9% 0.021 166.113);

--color-emerald-100: oklch(95% 0.052 163.051);

--color-emerald-200: oklch(90.5% 0.093 164.15);

--color-emerald-300: oklch(84.5% 0.143 164.978);

--color-emerald-400: oklch(76.5% 0.177 163.223);

--color-emerald-500: oklch(69.6% 0.17 162.48);

--color-emerald-600: oklch(59.6% 0.145 163.225);

--color-emerald-700: oklch(50.8% 0.118 165.612);

--color-emerald-800: oklch(43.2% 0.095 166.913);

--color-emerald-900: oklch(37.8% 0.077 168.94);

--color-emerald-950: oklch(26.2% 0.051 172.552);

--color-teal-50: oklch(98.4% 0.014 180.72);

--color-teal-100: oklch(95.3% 0.051 180.801);

--color-teal-200: oklch(91% 0.096 180.426);

--color-teal-300: oklch(85.5% 0.138 181.071);

--color-teal-400: oklch(77.7% 0.152 181.912);

--color-teal-500: oklch(70.4% 0.14 182.503);

--color-teal-600: oklch(60% 0.118 184.704);

--color-teal-700: oklch(51.1% 0.096 186.391);

--color-teal-800: oklch(43.7% 0.078 188.216);

--color-teal-900: oklch(38.6% 0.063 188.416);

--color-teal-950: oklch(27.7% 0.046 192.524);

--color-cyan-50: oklch(98.4% 0.019 200.873);

--color-cyan-100: oklch(95.6% 0.045 203.388);

--color-cyan-200: oklch(91.7% 0.08 205.041);

--color-cyan-300: oklch(86.5% 0.127 207.078);

--color-cyan-400: oklch(78.9% 0.154 211.53);

--color-cyan-500: oklch(71.5% 0.143 215.221);

--color-cyan-600: oklch(60.9% 0.126 221.723);

--color-cyan-700: oklch(52% 0.105 223.128);

--color-cyan-800: oklch(45% 0.085 224.283);

--color-cyan-900: oklch(39.8% 0.07 227.392);

--color-cyan-950: oklch(30.2% 0.056 229.695);

--color-sky-50: oklch(97.7% 0.013 236.62);

--color-sky-100: oklch(95.1% 0.026 236.824);

--color-sky-200: oklch(90.1% 0.058 230.902);

--color-sky-300: oklch(82.8% 0.111 230.318);

--color-sky-400: oklch(74.6% 0.16 232.661);

--color-sky-500: oklch(68.5% 0.169 237.323);

--color-sky-600: oklch(58.8% 0.158 241.966);

--color-sky-700: oklch(50% 0.134 242.749);

--color-sky-800: oklch(44.3% 0.11 240.79);

--color-sky-900: oklch(39.1% 0.09 240.876);

--color-sky-950: oklch(29.3% 0.066 243.157);

--color-blue-50: oklch(97% 0.014 254.604);

--color-blue-100: oklch(93.2% 0.032 255.585);

--color-blue-200: oklch(88.2% 0.059 254.128);

--color-blue-300: oklch(80.9% 0.105 251.813);

--color-blue-400: oklch(70.7% 0.165 254.624);

--color-blue-500: oklch(62.3% 0.214 259.815);

--color-blue-600: oklch(54.6% 0.245 262.881);

--color-blue-700: oklch(48.8% 0.243 264.376);

--color-blue-800: oklch(42.4% 0.199 265.638);

--color-blue-900: oklch(37.9% 0.146 265.522);

--color-blue-950: oklch(28.2% 0.091 267.935);

--color-indigo-50: oklch(96.2% 0.018 272.314);

--color-indigo-100: oklch(93% 0.034 272.788);

--color-indigo-200: oklch(87% 0.065 274.039);

--color-indigo-300: oklch(78.5% 0.115 274.713);

--color-indigo-400: oklch(67.3% 0.182 276.935);

--color-indigo-500: oklch(58.5% 0.233 277.117);

--color-indigo-600: oklch(51.1% 0.262 276.966);

--color-indigo-700: oklch(45.7% 0.24 277.023);

--color-indigo-800: oklch(39.8% 0.195 277.366);

--color-indigo-900: oklch(35.9% 0.144 278.697);

--color-indigo-950: oklch(25.7% 0.09 281.288);

--color-violet-50: oklch(96.9% 0.016 293.756);

--color-violet-100: oklch(94.3% 0.029 294.588);

--color-violet-200: oklch(89.4% 0.057 293.283);

--color-violet-300: oklch(81.1% 0.111 293.571);

--color-violet-400: oklch(70.2% 0.183 293.541);

--color-violet-500: oklch(60.6% 0.25 292.717);

--color-violet-600: oklch(54.1% 0.281 293.009);

--color-violet-700: oklch(49.1% 0.27 292.581);

--color-violet-800: oklch(43.2% 0.232 292.759);

--color-violet-900: oklch(38% 0.189 293.745);

--color-violet-950: oklch(28.3% 0.141 291.089);

--color-purple-50: oklch(97.7% 0.014 308.299);

--color-purple-100: oklch(94.6% 0.033 307.174);

--color-purple-200: oklch(90.2% 0.063 306.703);

--color-purple-300: oklch(82.7% 0.119 306.383);

--color-purple-400: oklch(71.4% 0.203 305.504);

--color-purple-500: oklch(62.7% 0.265 303.9);

--color-purple-600: oklch(55.8% 0.288 302.321);

--color-purple-700: oklch(49.6% 0.265 301.924);

--color-purple-800: oklch(43.8% 0.218 303.724);

--color-purple-900: oklch(38.1% 0.176 304.987);

--color-purple-950: oklch(29.1% 0.149 302.717);

--color-fuchsia-50: oklch(97.7% 0.017 320.058);

--color-fuchsia-100: oklch(95.2% 0.037 318.852);

--color-fuchsia-200: oklch(90.3% 0.076 319.62);

--color-fuchsia-300: oklch(83.3% 0.145 321.434);

--color-fuchsia-400: oklch(74% 0.238 322.16);

--color-fuchsia-500: oklch(66.7% 0.295 322.15);

--color-fuchsia-600: oklch(59.1% 0.293 322.896);

--color-fuchsia-700: oklch(51.8% 0.253 323.949);

--color-fuchsia-800: oklch(45.2% 0.211 324.591);

--color-fuchsia-900: oklch(40.1% 0.17 325.612);

--color-fuchsia-950: oklch(29.3% 0.136 325.661);

--color-pink-50: oklch(97.1% 0.014 343.198);

--color-pink-100: oklch(94.8% 0.028 342.258);

--color-pink-200: oklch(89.9% 0.061 343.231);

--color-pink-300: oklch(82.3% 0.12 346.018);

--color-pink-400: oklch(71.8% 0.202 349.761);

--color-pink-500: oklch(65.6% 0.241 354.308);

--color-pink-600: oklch(59.2% 0.249 0.584);

--color-pink-700: oklch(52.5% 0.223 3.958);

--color-pink-800: oklch(45.9% 0.187 3.815);

--color-pink-900: oklch(40.8% 0.153 2.432);

--color-pink-950: oklch(28.4% 0.109 3.907);

--color-rose-50: oklch(96.9% 0.015 12.422);

--color-rose-100: oklch(94.1% 0.03 12.58);

--color-rose-200: oklch(89.2% 0.058 10.001);

--color-rose-300: oklch(81% 0.117 11.638);

--color-rose-400: oklch(71.2% 0.194 13.428);

--color-rose-500: oklch(64.5% 0.246 16.439);

--color-rose-600: oklch(58.6% 0.253 17.585);

--color-rose-700: oklch(51.4% 0.222 16.935);

--color-rose-800: oklch(45.5% 0.188 13.697);

--color-rose-900: oklch(41% 0.159 10.272);

--color-rose-950: oklch(27.1% 0.105 12.094);

--color-slate-50: oklch(98.4% 0.003 247.858);

--color-slate-100: oklch(96.8% 0.007 247.896);

--color-slate-200: oklch(92.9% 0.013 255.508);

--color-slate-300: oklch(86.9% 0.022 252.894);

--color-slate-400: oklch(70.4% 0.04 256.788);

--color-slate-500: oklch(55.4% 0.046 257.417);

--color-slate-600: oklch(44.6% 0.043 257.281);

--color-slate-700: oklch(37.2% 0.044 257.287);

--color-slate-800: oklch(27.9% 0.041 260.031);

--color-slate-900: oklch(20.8% 0.042 265.755);

--color-slate-950: oklch(12.9% 0.042 264.695);

--color-gray-50: oklch(98.5% 0.002 247.839);

--color-gray-100: oklch(96.7% 0.003 264.542);

--color-gray-200: oklch(92.8% 0.006 264.531);

--color-gray-300: oklch(87.2% 0.01 258.338);

--color-gray-400: oklch(70.7% 0.022 261.325);

--color-gray-500: oklch(55.1% 0.027 264.364);

--color-gray-600: oklch(44.6% 0.03 256.802);

--color-gray-700: oklch(37.3% 0.034 259.733);

--color-gray-800: oklch(27.8% 0.033 256.848);

--color-gray-900: oklch(21% 0.034 264.665);

--color-gray-950: oklch(13% 0.028 261.692);

--color-zinc-50: oklch(98.5% 0 0);

--color-zinc-100: oklch(96.7% 0.001 286.375);

--color-zinc-200: oklch(92% 0.004 286.32);

--color-zinc-300: oklch(87.1% 0.006 286.286);

--color-zinc-400: oklch(70.5% 0.015 286.067);

--color-zinc-500: oklch(55.2% 0.016 285.938);

--color-zinc-600: oklch(44.2% 0.017 285.786);

--color-zinc-700: oklch(37% 0.013 285.805);

--color-zinc-800: oklch(27.4% 0.006 286.033);

--color-zinc-900: oklch(21% 0.006 285.885);

--color-zinc-950: oklch(14.1% 0.005 285.823);

--color-neutral-50: oklch(98.5% 0 0);

--color-neutral-100: oklch(97% 0 0);

--color-neutral-200: oklch(92.2% 0 0);

--color-neutral-300: oklch(87% 0 0);

--color-neutral-400: oklch(70.8% 0 0);

--color-neutral-500: oklch(55.6% 0 0);

--color-neutral-600: oklch(43.9% 0 0);

--color-neutral-700: oklch(37.1% 0 0);

--color-neutral-800: oklch(26.9% 0 0);

--color-neutral-900: oklch(20.5% 0 0);

--color-neutral-950: oklch(14.5% 0 0);

--color-stone-50: oklch(98.5% 0.001 106.423);

--color-stone-100: oklch(97% 0.001 106.424);

--color-stone-200: oklch(92.3% 0.003 48.717);

--color-stone-300: oklch(86.9% 0.005 56.366);

--color-stone-400: oklch(70.9% 0.01 56.259);

--color-stone-500: oklch(55.3% 0.013 58.071);

--color-stone-600: oklch(44.4% 0.011 73.639);

--color-stone-700: oklch(37.4% 0.01 67.558);

--color-stone-800: oklch(26.8% 0.007 34.298);

--color-stone-900: oklch(21.6% 0.006 56.043);

--color-stone-950: oklch(14.7% 0.004 49.25);

--color-black: #000;

--color-white: #fff;

--spacing: 0.25rem;

--breakpoint-sm: 40rem;

--breakpoint-md: 48rem;

--breakpoint-lg: 64rem;

--breakpoint-xl: 80rem;

--breakpoint-2xl: 96rem;

--container-3xs: 16rem;

--container-2xs: 18rem;

--container-xs: 20rem;

--container-sm: 24rem;

--container-md: 28rem;

--container-lg: 32rem;

--container-xl: 36rem;

--container-2xl: 42rem;

--container-3xl: 48rem;

--container-4xl: 56rem;

--container-5xl: 64rem;

--container-6xl: 72rem;

--container-7xl: 80rem;

--text-xs: 0.75rem;

--text-xs--line-height: calc(1 / 0.75);

--text-sm: 0.875rem;

--text-sm--line-height: calc(1.25 / 0.875);

--text-base: 1rem;

--text-base--line-height: calc(1.5 / 1);

--text-lg: 1.125rem;

--text-lg--line-height: calc(1.75 / 1.125);

--text-xl: 1.25rem;

--text-xl--line-height: calc(1.75 / 1.25);

--text-2xl: 1.5rem;

--text-2xl--line-height: calc(2 / 1.5);

--text-3xl: 1.875rem;

--text-3xl--line-height: calc(2.25 / 1.875);

--text-4xl: 2.25rem;

--text-4xl--line-height: calc(2.5 / 2.25);

--text-5xl: 3rem;

--text-5xl--line-height: 1;

--text-6xl: 3.75rem;

--text-6xl--line-height: 1;

--text-7xl: 4.5rem;

--text-7xl--line-height: 1;

--text-8xl: 6rem;

--text-8xl--line-height: 1;

--text-9xl: 8rem;

--text-9xl--line-height: 1;

--font-weight-thin: 100;

--font-weight-extralight: 200;

--font-weight-light: 300;

--font-weight-normal: 400;

--font-weight-medium: 500;

--font-weight-semibold: 600;

--font-weight-bold: 700;

--font-weight-extrabold: 800;

--font-weight-black: 900;

--tracking-tighter: -0.05em;

--tracking-tight: -0.025em;

--tracking-normal: 0em;

--tracking-wide: 0.025em;

--tracking-wider: 0.05em;

--tracking-widest: 0.1em;

--leading-tight: 1.25;

--leading-snug: 1.375;

--leading-normal: 1.5;

--leading-relaxed: 1.625;

--leading-loose: 2;

--radius-xs: 0.125rem;

--radius-sm: 0.25rem;

--radius-md: 0.375rem;

--radius-lg: 0.5rem;

--radius-xl: 0.75rem;

--radius-2xl: 1rem;

--radius-3xl: 1.5rem;

--radius-4xl: 2rem;

--shadow-2xs: 0 1px rgb(0 0 0 / 0.05);

--shadow-xs: 0 1px 2px 0 rgb(0 0 0 / 0.05);

--shadow-sm: 0 1px 3px 0 rgb(0 0 0 / 0.1), 0 1px 2px -1px rgb(0 0 0 / 0.1);

--shadow-md: 0 4px 6px -1px rgb(0 0 0 / 0.1), 0 2px 4px -2px rgb(0 0 0 / 0.1);

--shadow-lg: 0 10px 15px -3px rgb(0 0 0 / 0.1), 0 4px 6px -4px rgb(0 0 0 / 0.1);

--shadow-xl: 0 20px 25px -5px rgb(0 0 0 / 0.1), 0 8px 10px -6px rgb(0 0 0 / 0.1);

--shadow-2xl: 0 25px 50px -12px rgb(0 0 0 / 0.25);

--inset-shadow-2xs: inset 0 1px rgb(0 0 0 / 0.05);

--inset-shadow-xs: inset 0 1px 1px rgb(0 0 0 / 0.05);

--inset-shadow-sm: inset 0 2px 4px rgb(0 0 0 / 0.05);

--drop-shadow-xs: 0 1px 1px rgb(0 0 0 / 0.05);

--drop-shadow-sm: 0 1px 2px rgb(0 0 0 / 0.15);

--drop-shadow-md: 0 3px 3px rgb(0 0 0 / 0.12);

--drop-shadow-lg: 0 4px 4px rgb(0 0 0 / 0.15);

--drop-shadow-xl: 0 9px 7px rgb(0 0 0 / 0.1);

--drop-shadow-2xl: 0 25px 25px rgb(0 0 0 / 0.15);

--text-shadow-2xs: 0px 1px 0px rgb(0 0 0 / 0.15);

--text-shadow-xs: 0px 1px 1px rgb(0 0 0 / 0.2);

--text-shadow-sm:

0px 1px 0px rgb(0 0 0 / 0.075), 0px 1px 1px rgb(0 0 0 / 0.075), 0px 2px 2px rgb(0 0 0 / 0.075);

--text-shadow-md:

0px 1px 1px rgb(0 0 0 / 0.1), 0px 1px 2px rgb(0 0 0 / 0.1), 0px 2px 4px rgb(0 0 0 / 0.1);

--text-shadow-lg:

0px 1px 2px rgb(0 0 0 / 0.1), 0px 3px 2px rgb(0 0 0 / 0.1), 0px 4px 8px rgb(0 0 0 / 0.1);

--ease-in: cubic-bezier(0.4, 0, 1, 1);

--ease-out: cubic-bezier(0, 0, 0.2, 1);

--ease-in-out: cubic-bezier(0.4, 0, 0.2, 1);

--animate-spin: spin 1s linear infinite;

--animate-ping: ping 1s cubic-bezier(0, 0, 0.2, 1) infinite;

--animate-pulse: pulse 2s cubic-bezier(0.4, 0, 0.6, 1) infinite;

--animate-bounce: bounce 1s infinite;

@keyframes spin {

to {

transform: rotate(360deg);

}

}

@keyframes ping {

75%,

100% {

transform: scale(2);

opacity: 0;

}

}

@keyframes pulse {

50% {

opacity: 0.5;

}

}

@keyframes bounce {

0%,

100% {

transform: translateY(-25%);

animation-timing-function: cubic-bezier(0.8, 0, 1, 1);

}

50% {

transform: none;

animation-timing-function: cubic-bezier(0, 0, 0.2, 1);

}

}

--blur-xs: 4px;

--blur-sm: 8px;

--blur-md: 12px;

--blur-lg: 16px;

--blur-xl: 24px;

--blur-2xl: 40px;

--blur-3xl: 64px;

--perspective-dramatic: 100px;

--perspective-near: 300px;

--perspective-normal: 500px;

--perspective-midrange: 800px;

--perspective-distant: 1200px;

--aspect-video: 16 / 9;

--default-transition-duration: 150ms;

--default-transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1);

--default-font-family: --theme(--font-sans, initial);

--default-font-feature-settings: --theme(--font-sans--font-feature-settings, initial);

--default-font-variation-settings: --theme(--font-sans--font-variation-settings, initial);

--default-mono-font-family: --theme(--font-mono, initial);

--default-mono-font-feature-settings: --theme(--font-mono--font-feature-settings, initial);

--default-mono-font-variation-settings: --theme(--font-mono--font-variation-settings, initial);

}

/\* Deprecated \*/

@theme default inline reference {

--blur: 8px;

--shadow: 0 1px 3px 0 rgb(0 0 0 / 0.1), 0 1px 2px -1px rgb(0 0 0 / 0.1);

--shadow-inner: inset 0 2px 4px 0 rgb(0 0 0 / 0.05);

--drop-shadow: 0 1px 2px rgb(0 0 0 / 0.1), 0 1px 1px rgb(0 0 0 / 0.06);

--radius: 0.25rem;

--max-width-prose: 65ch;

}

@tailwind utilities;

<script src="tslib.es6.js"></script>

<script src="tslib.js"></script>